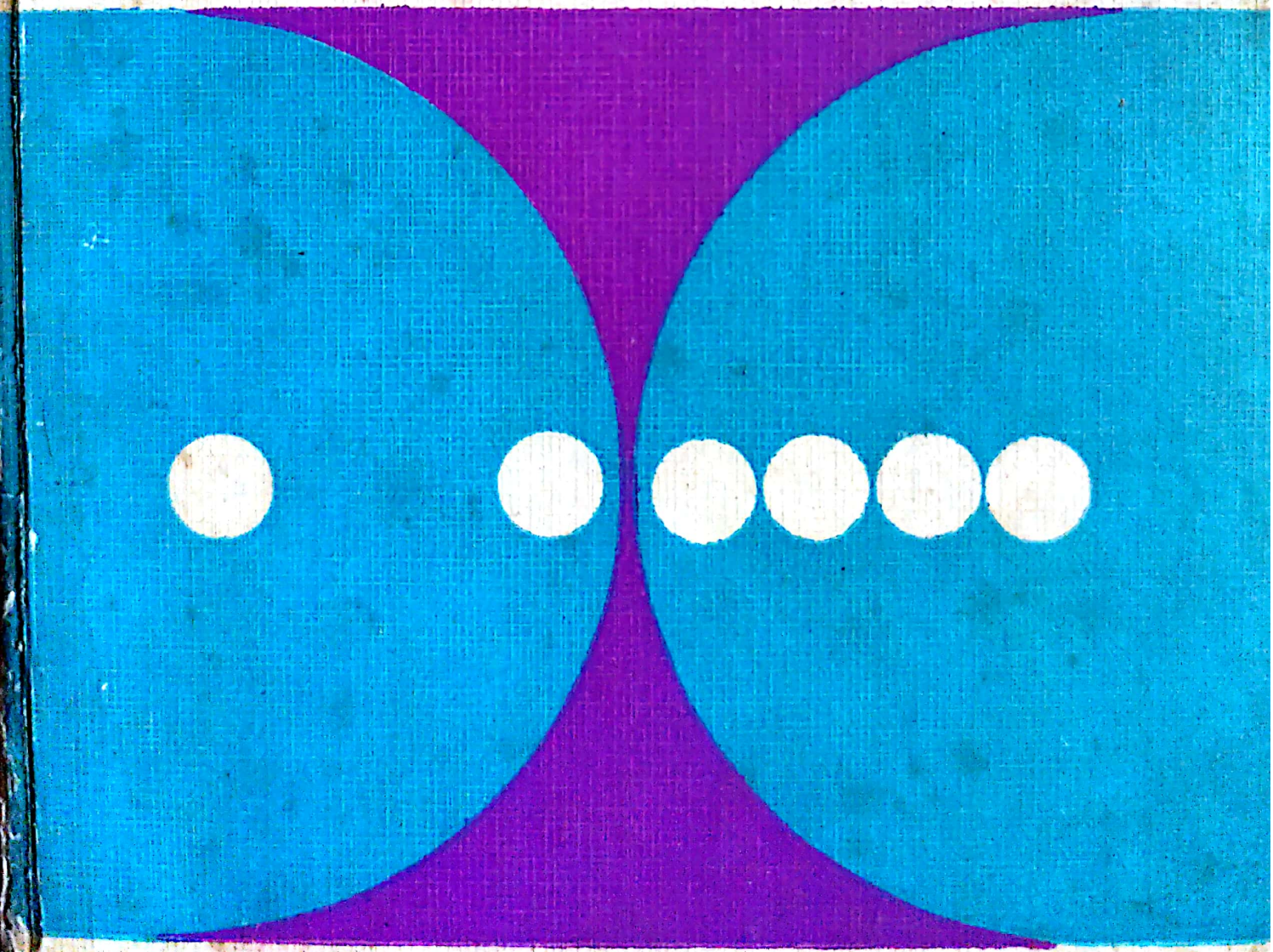
GHEORGHE MIHOC ● ANTON ȘTEFĂNESCU

PROGRAMAREA MATEMATICĂ

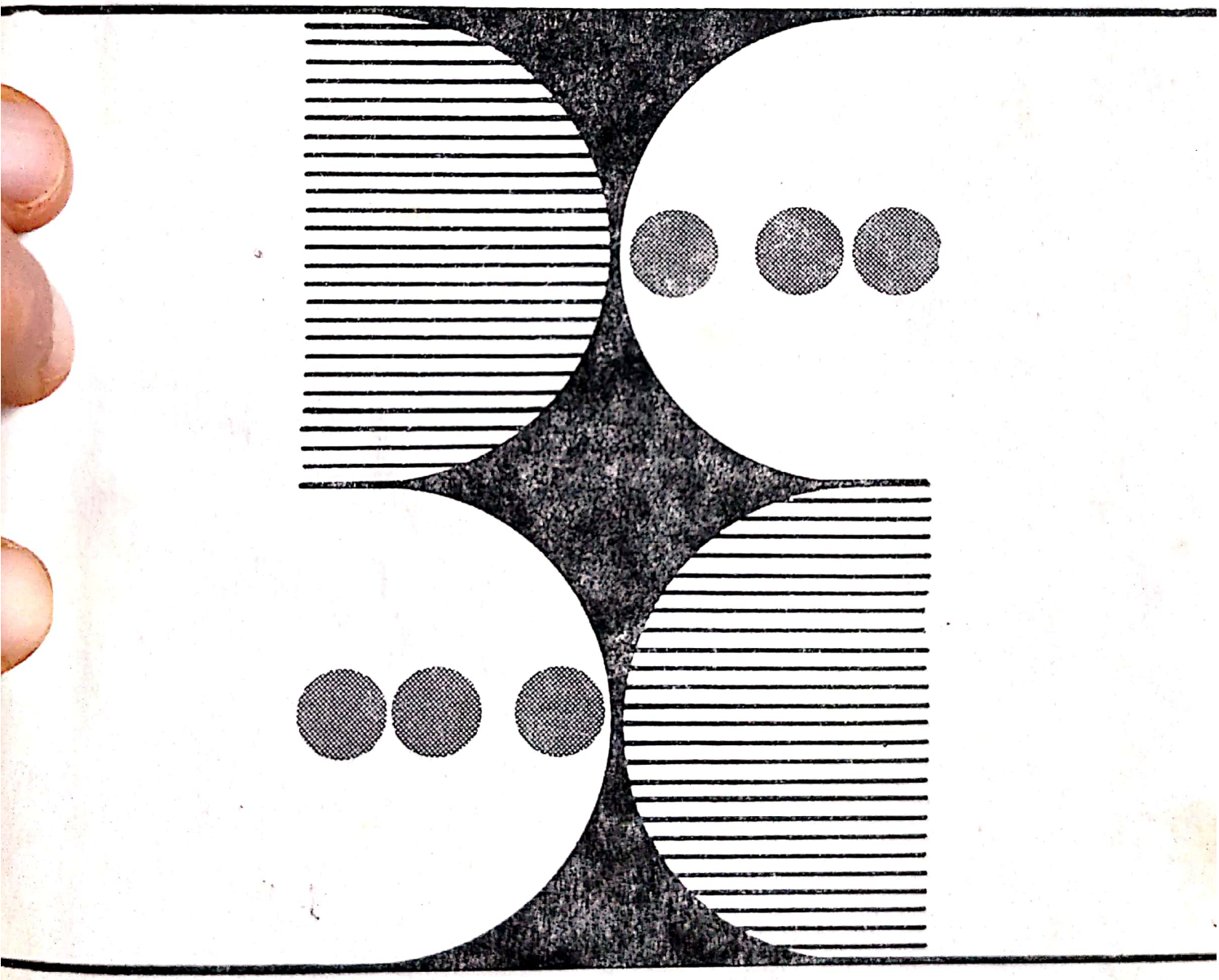


EDITURA DIDACTICĂ ȘI PEDAGOGICĂ
BUCUREȘTI

Scanned with OKEN Scanner

MINISTERUL EDUCAȚIEI ȘI ÎNVĂȚĂMÂNTULUI
Acad. GHEORGHE MIHOC
ANTON ȘTEFĂNESCU

PROGRAMAREA MATEMATICĂ



EDITURA DIDACTICĂ ȘI PEDAGOGICĂ
București – 1973

Scanned with OKEN Scanner

PREFAȚĂ

Probleme vizînd determinarea soluției optime în o serie de domenii ale științei au trezit interesul cercetătorilor din diferite epoci. În secolele 17 și 18 au fost elaborate o serie de procedee de optimizare, aplicabile unor probleme de natură geometrică sau de fizică.

În ultimul timp, mai precis în ultimele decenii ale secolului nostru, s-au impus atenției oamenilor de știință noi și numeroase probleme de optimizare în industrie, în planificarea economiei, în domeniul militar, etc., probleme a căror rezolvare reclamă folosirea unui aparat matematic foarte dezvoltat.

Așa se face că în urma difuzării metodelor științifice de tratare a fenomenelor dintr-o serie de domenii ale vieții economice și sociale, au apărut și se dezvoltă necontenit noi metode matematice de rezolvare a problemelor de optimizare.

O clasă, bine definită, a problemelor de acest tip, foarte importantă din punct de vedere aplicativ, o constituie cea a problemelor de programare matematică.

O problemă de programare matematică poate fi formulată astfel: se urmărește determinarea valorilor unor variabile $x_1, x_2, \ldots, x_n$, obligate să satisfacă un număr dat de condiții (de regulă exprimate prin egalități sau inegalități):

$$g_i(x_1, x_2, \ldots, x_n) \{\geqslant, =, \leqslant\} 0 \tag{1}$$

și care, în plus, să maximizeze sau să minimizeze o funcție:

$$f(x_1, x_2, \ldots, x_n).$$

Uneori, asupra variabilelor se pun și alte condiții cu o formă deosebită de (1); de exemplu condiții care limitează valorile variabilelor la mulțimea numerelor întregi.

Scanned with OKEN Scanner

O astfel de formulare apare întotdeauna ca efect al modelării unui fenomen întîlnit în practică, de cele mai multe ori, unul de natură economică.

Acesta este motivul pentru care o serie de termeni de factură economică s-au impus în teoria matematică a programării matematice. Astfel, (1) poartă numele de sistem de restricţii, funcţia f o numim, curent, funcţie de eficienţă.

Iniţial, modelele programării matematice s-au studiat în cadrul unor teorii matematice deja existente, de cele mai multe ori, însă, ele au reclamat crearea unui aparat matematic nou.

Deşi toate problemele programării matematice au un aspect formal asemănător, totuşi nu se poate vorbi de o teorie matematică unitară. Aceasta, deoarece dincolo de asemănarea formală există deosebiri profunde de ordin teoretic.

Din această cauză, problemele programării matematice se tratează separat, împărţite în clase distincte, în funcţie de instrumentul matematic impus de particularităţile structurii lor specifice.

Clasa problemelor de programare liniară este caracterizată prin expresii liniare, atît pentru toate funcţiile g_i din sistemul de restricţii (1), cît şi pentru funcţia de eficienţă f.

Primele probleme de acest gen, legate de planificarea în economia socialistă, au fost elaborate de matematicianul sovietic L. V. Kantorovici. Forma matematică generală a problemelor programării liniare, aşa cum este acceptată astăzi unanim, a fost dată de G. Dantzig în 1947, care a propus, totodată şi un procedeu eficient de rezolvare — metoda simplex.

La ora actuală, se poate vorbi despre un mod unitar de abordare al tuturor problemelor de programare liniară. Diversele metode de rezolvare existente au ca raţiune sporirea rapidităţii în calcule, în funcţie de particularităţile fiecărei probleme, avînd în vedere, în special, utilizarea calculatoarelor.

Cercetarea în problemele de programare neliniară s-a lovit de dificultăţi matematice mult mai însemnate.

Deşi, fără îndoială, s-au înregistrat progrese enorme şi în acest domeniu, totuşi nu se poate vorbi despre o teorie completă a programării neliniare. Cele mai importante rezultate teoretice obţinute, precum şi metodele de rezolvare, au o aplicabilitate restrînsă la un grup de probleme cu o structură particulară

Scanned with OKEN Scanner

Un tip de probleme de programare matematică care este deobicei tratat ca aparţinînd unei clase distincte, îl constituie cel în care apar restricţii speciale asupra variabilelor $x_1, x_2, \ldots, x_n$ care le obligă pe acestea să ia doar valori întregi. Pentru problemele aparţinînd acestei clase — numită programare discretă —, au fost elaborate, în unele cazuri particulare, algoritme speciale de rezolvare.

De o foarte mare atenţie se bucură şi problemele programării matematice în care, pe lîngă variabilele de decizie $x_1, x_2, \ldots, x_n$, apare şi variabilitatea unor coeficienţi care definesc sistemul de restricţii (1), sau funcţia de eficienţă f.

În cazul cînd această variabilitate are caracter deterministic, adică pentru clasa problemelor de programare parametrică, metodele de rezolvare sînt obţinute prin adaptarea metodelor generale utilizate în programarea liniară sau neliniară.

Mult mai complicată este însă abordarea problemelor de programare stochastică, în care variabilitatea coeficienţilor are un caracter aleator.

În sfîrşit, de un interes deosebit se bucură o tehnică specială de rezolvare a unor probleme de optimizare, elaborată relativ recent de R. Bellman, cunoscută sub numele de programare dinamică.

În prezenta lucrare ne-am propus să prezentăm principalele tipuri de probleme ale programării, de o manieră accesibilă şi celor fără o pregătire matematică specială.

Întrucît aparatul matematic utilizat în programarea liniară este destul de rudimentar, am insistat mai mult asupra acestui capitol.

Pentru celelalte clase de probleme, pe măsură ce dificultăţile de ordin matematic sporesc, ne-am mulţumit să schiţăm doar cadrul teoretic în care se plasează problema respectivă, insistînd mai mult asupra metodelor efective de rezolvare.

Am dat multe exemple cu dublu scop: de a-l informa pe cititor asupra domeniilor practice în care îşi găsesc aplicaţii problemele programării matematice, precum şi pentru a susţine cu date numerice metodele de rezolvare prezentate.

AUTORII

Scanned with OKEN Scanner

Capitolul I

NOȚIUNI MATEMATICE DE BAZĂ

Plecînd de la premiza că, în general sînt cunoscute proprietățile matricelor, determinanților și ale spațiilor vectoriale, acestui capitol îi revine rolul de a fixa terminologia și notațiile utilizate în această lucrare. Totodată vor fi enunțate sumar, unele proprietăți și rezultate privind mulțimile și funcțiile convexe, formele liniare și pătratice și sistemele de inecuații liniare.

§ 1. ALGEBRĂ MATRICEALĂ

1.1. Matrice. Operații cu matrice

Vom numi matrice de ordin $m \times n$, un tablou ordonat de mn numere

$$A = \begin{Vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{Vmatrix}.$$

Numerele a_{ij}(i putînd lua valorile 1, 2, ..., m, iar j valorile 1, 2, ..., n), se numesc elementele matricei.

Punînd în evidență elementul general a_{ij}(cel ce se găsește plasat pe linia i și coloana j), se folosește obișnuit și notația

$$A = \|a_{ij}\|_{\substack{i=1,2,\dots,m \\ j=1,2,\dots,n}}$$

Scanned with OKEN Scanner

Cînd $m = 1$, forma particulară a matricei $A = ||a_{11}\ a_{12}\ \ldots\ a_{1n}||$ poartă numele de *matrice* (sau *vector*) — *linie.*

Cînd $n = 1$,

$$A = \left\| \begin{matrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{matrix} \right\|$$

se va numi *matrice* (*vector*)-*coloană.*

Două matrice A şi B sînt egale dacă sînt de acelaşi ordin $(m \times n)$ şi dacă $a_{ij} = b_{ij}$ pentru $i = 1, 2, \ldots, m, j = 1, 2, \ldots, n.$

În mulţimea matricelor de acelaşi ordin $(m \times n)$ se introduc două operaţii algebrice: adunarea şi înmulţirea cu un scalar.

Adunarea matricelor este operaţia care asociază oricăror două matrice $A = ||a_{ij}||_{\substack{i=1,2,\ldots,m \\ j=1,2,\ldots,n}}$, $B = ||b_{ij}||_{\substack{i=1,2,\ldots,m \\ j=1,2,\ldots,n}}$ matricea C, de acelaşi ordin, ale cărei elemente c_{ij} sînt definite de relaţia

$$c_{ij} = a_{ij} + b_{ij}; \quad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n.$$

Înmulţirea cu un scalar, asociază matricei $A = ||a_{ij}||_{\substack{i=1,2,\ldots,m \\ j=1,2,\ldots,n}}$ şi numărului λ, matricea λA, de acelaşi ordin cu A, formată din elementele λa_{ij}; $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n.$

Exemplu

$$A = \left\| \begin{matrix} 1 & 3 & -1 \\ 0 & 2 & 1 \end{matrix} \right\|, \quad B = \left\| \begin{matrix} -2 & 1 & 4 \\ 3 & -1 & 2 \end{matrix} \right\|, \quad \lambda = 5;$$

atunci

$$\lambda A + B = \left\| \begin{matrix} 5 & 15 & -5 \\ 0 & 10 & 5 \end{matrix} \right\| + \left\| \begin{matrix} -2 & 1 & 4 \\ 3 & -1 & 2 \end{matrix} \right\| = \left\| \begin{matrix} 3 & 16 & -1 \\ 3 & 9 & 7 \end{matrix} \right\|.$$

Vom nota cu O matricea de ordin $m \times n$, cu toate elementele egale cu 0, adică

$$O = \left\| \begin{matrix} 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 0 \end{matrix} \right\|.$$

Scanned with OKEN Scanner

Atunci, între matricele de acelaşi ordin, constituite din numere reale, se poate defini o relaţie de ordine şi anume

$$A \geqslant B \text{ dacă } A - B \geqslant 0$$

ultima relaţie implicînd faptul că toate elementele $a_{ij} - b_{ij}$ ale matricii $A - B$ sînt nenegative.

Dacă $A = ||a_{ij}||_{\substack{i=1,2,\ldots,m \\ j=1,2,\ldots,n}}$ este o matrice de ordin $(m \times n)$ şi $B = ||b_{ij}||_{\substack{i=1,2,\ldots,n \\ j=1,2,\ldots,p}}$, de tipul $(n \times p)$ se defineşte matricea produs $A \cdot B$ ca fiind o matrice C de ordinul $(m \times p)$, ale cărei elemente c_{ij} sînt

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}, \quad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, p,$$

relaţie, potrivit căreia, elementul de pe linia i şi coloana j a matricei produs, se obţine sumînd produsele elementelor corespunzătoare ale liniei i a matricei A şi coloanei j a matricei B.

Aşa cum este definit, produsul se poate calcula numai în cazul cînd numărul de coloane ale primului factor coincide cu numărul de linii al celui de al doilea.

De notat că produsul matricelor nu posedă proprietatea de comutativitate; de fapt, dintre produsele $A \cdot B$ şi $B \cdot A$ de cele mai multe ori nu poate fi efectuat decît unul singur.

Exemple

a)
$$A = \begin{Vmatrix} 2 & 1 \\ 3 & 5 \\ 4 & 0 \end{Vmatrix}, \quad B = \begin{Vmatrix} 1 & 0 & -1 \\ 2 & 3 & 1 \\ 1 & 0 & -2 \\ 4 & 5 & 1 \end{Vmatrix}.$$

Are sens produsul $B \cdot A$ deoarece numărul coloanelor lui B coincide cu numărul liniilor lui A.

Matricea produs va fi de ordinul (4×2) şi anume

$$B \cdot A = \begin{Vmatrix} -2 & 1 \\ 17 & 17 \\ -6 & 1 \\ 27 & 29 \end{Vmatrix}$$

Scanned with OKEN Scanner

b)

$$A = \begin{Vmatrix} 2 & 3 & 1 \\ 1 & -1 & 0 \end{Vmatrix},\ B = \begin{Vmatrix} 1 \\ 3 \\ -1 \end{Vmatrix},\ AB = \begin{Vmatrix} 10 \\ -2 \end{Vmatrix};$$

c)

$$A = ||1 \quad -1 \quad 2||,\quad B = \begin{Vmatrix} 1 & 3 & -1 & 2 \\ 2 & 1 & 0 & 1 \\ 0 & 0 & -1 & 3 \end{Vmatrix},\ AB = ||-12 -37||$$

d)

$$A = ||1 \quad -1 \quad 2||,\quad B = \begin{Vmatrix} 1 \\ 3 \\ -1 \end{Vmatrix},$$ au sens AB şi BA.

Pentru o matrice dată $A = ||a_{ij}||_{\substack{i=1,2,\ldots,m \\ j=1,2,\ldots,n}}$, transpusa sa este matricea A' formată din A schimbînd liniile în coloane, astfel ca linia i din A devine coloana i în A'.

Deci $A' = ||a'_{ij}||_{\substack{i=1,2,\ldots,n \\ j=1,2,\ldots,m}}$; unde $a'_{ij} = a_{ji}$. Se observă că dacă A este de ordinul $m \times n$, atunci A' este de ordinul $(n \times m)$.

Dintre proprietăţile transpusei notăm

$$\begin{aligned} (A')' &= A, \\ (A + B)' &= A' + B', \\ (A \cdot B)' &= B' \cdot A'. \end{aligned}$$

1.2. Partiţionarea matricelor

Dacă dintr-o matrice A de ordinul $(m \times n)$ se elimină toate liniile şi coloanele cu excepţia unui număr de k linii $(k \leqslant m)$, şi respectiv, l coloane $(l \leqslant n)$, atunci cele $k \cdot l$ elemente rămase formează o matrice de ordinul $(k \times l)$ numită submatrice a lui A.

Exemplu.

$$A = \begin{Vmatrix} 1 & 0 & -1 & 3 & 5 \\ 2 & 7 & 2 & -3 & 1 \\ 0 & 0 & 2 & -1 & 1 \end{Vmatrix}.$$

Scanned with OKEN Scanner

Dacă se elimină linia 2 şi coloanele 1, 3 se obţine o submatrice de ordinul (2×3),

$$\begin{Vmatrix} 0 & 3 & 5 \\ 0 & -1 & 1 \end{Vmatrix}.$$

Din diferite raţiuni de calcul, uneori este necesar să se pună în evidenţă, în cadrul unei anumite matrice, o parte din elementele sale ordonate în una sau mai multe submatrice.

Un asemenea procedeu de împărţire a elementelor unei matrice în celule dreptunghiulare, ce formează submatrice ale sale, va fi numit partiţionare.

De exemplu fie

$$A = \begin{Vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \\ a_{51} & a_{52} & a_{53} \end{Vmatrix}.$$

Notînd

$$A_{11} = \begin{Vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{Vmatrix}, \quad A_{12} = \begin{Vmatrix} a_{13} \\ a_{23} \\ a_{33} \end{Vmatrix}$$

$$A_{21} = \begin{Vmatrix} a_{41} & a_{42} \\ a_{51} & a_{52} \end{Vmatrix}, \quad A_{22} = \begin{Vmatrix} a_{43} \\ a_{53} \end{Vmatrix}.$$

se obţine partiţionarea lui A,

$$A = \begin{Vmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{Vmatrix}.$$

Este evident faptul că partiţionarea matricei A se poate face în mai multe moduri.

Frecvent, procedeul de partiţionare este folosit pentru simplificarea efectuării unor operaţii, de exemplu înmulţirea. În această

Scanned with OKEN Scanner

situaţie produsul se face pe blocuri ; adică în construcţia matricei produs vor apărea produsele submatricelor celor două matrice factori.

Dacă

$$B = \left\| \begin{matrix} b_{11} & b_{12} & . & b_{13} & b_{14} \\ b_{21} & b_{22} & . & b_{23} & b_{24} \\ . & . & . & . & . \\ b_{31} & b_{32} & . & b_{33} & b_{34} \end{matrix} \right\|$$

este partiţionată în submatricele

$$B_{11} = \left\| \begin{matrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{matrix} \right\|, \quad B_{12} = \left\| \begin{matrix} b_{13} & b_{14} \\ b_{23} & b_{24} \end{matrix} \right\|,$$

$$B_{21} = \|b_{31} \quad b_{32}\|, \quad B_{22} = \|b_{33} \quad b_{34}\|$$

atunci produsul $A \cdot B$ se poate efectua după regula

$$\left\| \begin{matrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{matrix} \right\| \cdot \left\| \begin{matrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{matrix} \right\| = \left\| \begin{matrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{matrix} \right\|.$$

Se remarcă imediat că toate operaţiile înscrise în membrul doi au sens.

Rezultatul constituie totodată, o partiţionare a matricei produs $A \cdot B = C$ de forma

$$C = \left\| \begin{matrix} c_{11} & c_{12} & . & c_{13} & c_{14} \\ c_{21} & c_{22} & . & c_{23} & c_{24} \\ c_{31} & c_{32} & . & c_{33} & c_{34} \\ . & . & . & . & . \\ c_{41} & c_{42} & . & c_{43} & c_{44} \\ c_{51} & c_{52} & . & c_{53} & c_{54} \end{matrix} \right\| = \left\| \begin{matrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{matrix} \right\|$$

unde

$$C_{11} = A_{11}B_{11} + A_{12}B_{21}, \qquad C_{12} = A_{11}B_{12} + A_{12}B_{22},$$

$$C_{21} = A_{21}B_{11} + A_{22}B_{21}, \qquad C_{22} = A_{21}B_{12} + A_{22}B_{22}.$$

Scanned with OKEN Scanner

În particular să notăm cu $P_1, P_2, \ldots, P_n$ submatricele coloană obținute din matricea de ordin $m \times n$, $A = ||a_{ij}||_{\substack{i=1,2,\ldots,m \\ j=1,2,\ldots,n}}$

$$P_j = \left\| \begin{matrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{matrix} \right\|.$$

Dacă X este o matrice coloană de ordinul $(n \times 1)$, $X = \left\| \begin{matrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{matrix} \right\|$,

se poate efectua produsul AX sub forma

$$A \cdot X = ||P_1\, P_2\, P_3 \ldots P_n|| \cdot \left\| \begin{matrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{matrix} \right\| = P_1x_1 + P_2x_2 + P_3x_3 + \ldots + P_nx_n.$$

Exemple

$$A = \left\| \begin{array}{ccc|cc} 2 & -1 & 3 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ \hline 3 & 2 & -1 & 1 & 2 \\ \hline 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array} \right\|, \quad B = \left\| \begin{array}{cc} 1 & 3 \\ -1 & 0 \\ 2 & 2 \\ \hline 1 & 1 \\ 0 & -2 \end{array} \right\|$$

Este avantajos, pentru efectuarea produsului $A \cdot B$, să partiționăm matricele, în celulele reprezentate în tablou.

Punem deci

$$A = \left\| \begin{matrix} A_{11} & A_{12} \\ A_{21} & A_{22} \\ A_{31} & A_{32} \end{matrix} \right\|, \quad B = \left\| \begin{matrix} B_{11} \\ B_{21} \end{matrix} \right\|$$

Scanned with OKEN Scanner

unde

$$A_{11} = \begin{Vmatrix} 2 & -1 & 3 \\ 1 & 0 & -1 \end{Vmatrix}, \quad A_{12} = \begin{Vmatrix} 0 & 0 \\ 0 & 0 \end{Vmatrix},$$

$$A_{21} = \| 3 \quad 2 \quad -1 \|, \quad A_{22} = \| 1 \quad 2 \|,$$

$$A_{31} = \begin{Vmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{Vmatrix}, \quad A_{32} = \begin{Vmatrix} 1 & 0 \\ 0 & 1 \end{Vmatrix},$$

$$B_{11} = \begin{Vmatrix} 1 & 3 \\ -1 & 0 \\ 2 & 2 \end{Vmatrix}, \quad B_{21} = \begin{Vmatrix} 1 & 1 \\ 0 & -2 \end{Vmatrix}.$$

Pentru produsul AB folosim reprezentarea

$$AB = \begin{Vmatrix} A_{11}B_{11} + A_{12}B_{21} \\ A_{21}B_{11} + A_{22}B_{21} \\ A_{31}B_{11} + A_{32}B_{21} \end{Vmatrix}$$

şi cum produsele în care apar ca factor A_{12} şi A_{31} sînt nule, numărul calculelor succesive se reduce sensibil.

$$AB = \left\| \begin{array}{cc} 9 & 12 \\ -1 & 1 \\ \hline 0 & 4 \\ \hline 1 & 1 \\ 0 & -2 \end{array} \right\|.$$

Fie

$$A = \begin{Vmatrix} 5 & 2 & 1 & -1 & 0 \\ 2 & 0 & -1 & 2 & 1 \\ 1 & 2 & 1 & 0 & 0 \end{Vmatrix}, \quad B = \begin{Vmatrix} 1 \\ 0 \\ 1 \\ 2 \\ -1 \end{Vmatrix}.$$

Scanned with OKEN Scanner

Produsul $A \cdot B$, cu A partiţionată în cinci matrice coloană, se reduce la suma

$$1 \cdot \left\| \begin{matrix} 5 \\ 2 \\ 1 \end{matrix} \right\| + 0 \cdot \left\| \begin{matrix} 2 \\ 0 \\ 2 \end{matrix} \right\| + 1 \cdot \left\| \begin{matrix} 1 \\ -1 \\ 1 \end{matrix} \right\| + 2 \cdot \left\| \begin{matrix} -1 \\ 2 \\ 0 \end{matrix} \right\| - 1 \left\| \begin{matrix} 0 \\ 1 \\ 0 \end{matrix} \right\|.$$

1.3. Matrice pătrate. Determinanţi. Matrice nesingulare

Dacă numărul de linii coincide cu cel al coloanelor, matricea este pătrată.

O matrice A pătrată de ordinul $(n \times n)$ are forma

$$A = \left\| \begin{matrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{matrix} \right\|.$$

Punem în evidenţă unele tipuri particulare de matrice pătrate:

Matricea diagonală, este o matrice $D = \|d_{ij}\|_{\substack{i=1,2,\dots,n \\ j=1,2,\dots,n}}$ unde toţi $d_{ij} = 0$ cînd $i \neq j$, adică

$$D = \left\| \begin{matrix} d_{11} & 0 & 0 & \dots & 0 \\ 0 & d_{22} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & d_{nn} \end{matrix} \right\|.$$

Dacă în plus $d_{ii} = 1$ pentru $i = 1, 2, \dots, n$, matricea se va numi *matrice unitate*. Notaţia pe care o vom adopta va fi

$$E = \left\| \begin{matrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{matrix} \right\|.$$

Este necesar, pentru matricea unitate, să se specifice şi ordinul aşa că ori de cîte ori este nevoie, notaţia va fi completată

Scanned with OKEN Scanner

cu un indice. De exemplu E_n reprezintă matricea unitate de ordinul $(n \times n)$.

O matrice de tip superior triunghiular are proprietatea că elementele sale de sub diagonala principală sînt nule, adică $s_{ij} = 0$ dacă $i > j$.

Forma unei asemenea matrice este

$$S = \begin{Vmatrix} s_{11} & s_{12} & s_{13} & \dots & s_{1n} \\ 0 & s_{22} & s_{23} & \dots & s_{2n} \\ 0 & 0 & s_{33} & \dots & s_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & s_{nn} \end{Vmatrix}.$$

Analog, o matrice $S' = ||s'_{ij}||_{\substack{i=1,2,\dots,n \\ j=1,2,\dots,n}}$ este de tip inferior triunghiular dacă $s'_{ij} = 0$ pentru $i < j$,

$$S = \begin{Vmatrix} s_{11} & 0 & 0 & \dots & 0 \\ s_{21} & s_{22} & 0 & \dots & 0 \\ s_{31} & s_{32} & s_{33} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ s_{n1} & s_{n2} & s_{n3} & \dots & s_{nn} \end{Vmatrix}.$$

Determinantul unei matrice pătrate de ordinul $(n \times n)$ este numărul $|A|$ format din suma produselor de cîte n elemente ale matricei

$$|A| = \Sigma (\pm) a_{1j_1} a_{2j_2} a_{3j_3} \dots a_{nj_n}.$$

suma făcîndu-se după toate permutările posibile $j_1, j_2, \dots, j_n$ ale numerelor 1, 2, ..., n. Semnul unui termen al sumei este $+$ sau $-$ după cum permutarea este pară sau impară. Numărul n reprezintă ordinul determinantului. Conform acestei definiţii, determinantul ataşat matricei

$$A = \begin{Vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{Vmatrix},$$

Scanned with OKEN Scanner

este

$$|A| = a_{11}\ a_{22}\ a_{33} - a_{12}\ a_{21}a_{33} + a_{12}\ a_{23}\ a_{31} - a_{13}\ a_{22}\ a_{31} + \\ + a_{13}\ a_{21}\ a_{32} - a_{11}\ a_{23}\ a_{32} = (a_{11}\ a_{22}\ a_{33} + a_{12}\ a_{23}\ a_{31} + \\ + a_{13}\ a_{21}\ a_{32}) - (a_{12}\ a_{21}\ a_{33} + a_{13}\ a_{22}\ a_{31} + a_{11}\ a_{23}\ a_{32})$$

(regula lui Sarrus)

Prin definiţie, un minor de ordin k al unei matrice A de ordin $m \times n$ este determinantul unei submatrice de ordinul $(k \times k)$, a lui A (evident k trebuie să fie mai mic decît m şi n).

Minorul elementului a_{ij} al unei matrice pătrate $(n \times n)$, A este determinantul de ordin $n - 1$, ataşat submatricei obţinute prin eliminarea liniei i şi coloanei j.

Dacă minorul lui a_{ij} se multiplică cu $(-1)^{i+j}$, se obţine aşa numitul *complement algebric* al acestui element. În cele ce urmează complementul algebric al lui a_{ij} va fi notat cu A_{ij}. Pentru calculul determinanţilor se foloseşte un procedeu de dezvoltare după elementele unei linii sau ale unei coloane, procedeu care reduce calculul unui determinant de ordinul n la calculul a n determinanţi de ordinul $n - 1$.

Fie

$$A = \begin{Vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{i1} & a_{i2} & \dots & a_{in} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{Vmatrix};$$

dezvoltarea determinantului $|A|$ după linia i se face după regula

$$|A| = a_{i1}A_{i1} + a_{i2}A_{i2} + \dots + a_{in}A_{in} = \sum_{k=1}^{n} a_{ik}A_{ik}$$

De exemplu, pe baza acestei reguli, dezvoltînd după prima linie determinantul

$$|A| = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

Scanned with OKEN Scanner

se obține

$$|A| = a_{11} \cdot (-1)^2 \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} + a_{12}(-1)^3 \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{32} \end{vmatrix} +$$

$$+ a_{13} \cdot (-1)^4 \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} = a_{11}(a_{22}a_{33} - a_{23}a_{32}) -$$

$$- a_{12}(a_{21}a_{32} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31}) =$$

$$= a_{11}a_{22}a_{33} - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{32} + a_{12}a_{23}a_{31} +$$

$$+ a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31}.$$

Practic, pentru calculul unui determinant de ordin n, se poate aplica regula, succesiv, de un număr de ori, reducînd problema, în final, la calcularea unor determinanți de ordinul 2 sau 3.

O matrice pătrată A de ordin $(n \times n)$, se va numi *nesingulară* dacă $|A| \neq 0$.

În acest caz se demonstrează posibilitatea determinării unei matrice A^{-1} cu proprietatea

$$A^{-1} \cdot A = A \cdot A^{-1} = E.$$

Matricea A^{-1} se va numi *inversa* lui A.

Dăm mai jos procesul constructiv, în etape, pentru aflarea inversei unei matrice nesingulare.

Fie

$$A = \begin{Vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{Vmatrix} \text{ cu } |A| \neq 0.$$

Într-o primă etapă se construiește matricea transpusă

$$A' = \begin{Vmatrix} a_{11} & a_{21} & \dots & a_{n1} \\ a_{12} & a_{22} & \dots & a_{n2} \\ \dots & \dots & \dots & \dots \\ a_{1n} & a_{2n} & \dots & a_{nn} \end{Vmatrix}.$$

Scanned with OKEN Scanner

Înlocuind fiecare element al matricei A' cu complementul său algebric se obţine matricea adjunctă

$$A^* = \begin{Vmatrix} A_{11} & A_{21} & \dots & A_{n1} \\ A_{12} & A_{22} & \dots & A_{n2} \\ \dots & \dots & \dots & \dots \\ A_{1n} & A_{2n} & \dots & A_{nn} \end{Vmatrix}.$$

În sfîrşit, matricea inversă A^{-1} se obţine prin împărţire cu $|A|$ adică

$$A^{-1} = \frac{A^*}{|A|} = \begin{Vmatrix} \frac{A_{11}}{|A|} & \frac{A_{21}}{|A|} & \dots & \frac{A_{n1}}{|A|} \\ \frac{A_{12}}{|A|} & \frac{A_{22}}{|A|} & \dots & \frac{A_{n2}}{|A|} \\ \dots & \dots & \dots & \dots \\ \frac{A_{1n}}{|A|} & \frac{A_{2n}}{|A|} & \dots & \frac{A_{nn}}{|A|} \end{Vmatrix}.$$

Inversa unei matrice A este unică. Dintre celelalte proprietăţi importante citǎm

$$|A^{-1}| = \frac{1}{|A|}.$$

Dacă A^{-1} şi B^{-1} există atunci

$$(A \cdot B)^{-1} = B^{-1} \cdot A^{-1}$$

$$(A')^{-1} = (A^{-1})'.$$

Exemplu.

$$A = \begin{Vmatrix} 2 & 1 & 0 \\ -1 & 1 & 3 \\ 5 & -2 & 1 \end{Vmatrix}, \quad |A| = \begin{vmatrix} 2 & 1 & 0 \\ -1 & 1 & 3 \\ 5 & -2 & 1 \end{vmatrix} = 30,$$

$$A' = \begin{Vmatrix} 2 & -1 & 5 \\ 1 & 1 & -2 \\ 0 & 3 & 1 \end{Vmatrix}$$

Scanned with OKEN Scanner

$$A_{11}=\begin{vmatrix}1 & -2\\ 3 & 1\end{vmatrix}=7,\ A_{21}=-\begin{vmatrix}1 & -2\\ 0 & 1\end{vmatrix}=-1,$$

$$A_{31}=\begin{vmatrix}1 & 1\\ 0 & 3\end{vmatrix}=3,$$

$$A_{12}=-\begin{vmatrix}-1 & 5\\ 3 & 1\end{vmatrix}=+16,\ A_{22}=\begin{vmatrix}2 & 5\\ 0 & 1\end{vmatrix}=2,\ A_{32}=-\begin{vmatrix}2 & -1\\ 0 & 3\end{vmatrix}=$$
$$=-6,$$

$$A_{13}=\begin{vmatrix}-1 & 5\\ 1 & -2\end{vmatrix}=-3,\ A_{23}=-\begin{vmatrix}2 & 5\\ 1 & -2\end{vmatrix}=9,\ A_{33}=$$
$$=\begin{vmatrix}2 & -1\\ 1 & 1\end{vmatrix}=3,$$

$$A^*=\begin{Vmatrix}7 & -1 & 3\\ 16 & 2 & -6\\ -3 & 9 & 3\end{Vmatrix}\text{ și deci }A^{-1}=\begin{Vmatrix}7/30 & -1/30 & 3/30\\ 8/15 & 1/15 & -1/5\\ -1/10 & 3/10 & 1/10\end{Vmatrix},$$

1.4. Rangul unei matrice

Pentru o matrice A de ordinul $(m \times n)$ se definește rangul ca fiind ordinul celui mai mare minor nenul al său. Determinarea rangului joacă un rol foarte important în teoria sistemelor de ecuații liniare stînd la baza tuturor teoremelor de existență și a tuturor metodelor întrebuințate în rezolvare.

Din definiția rangului rezultă că el nu poate să depășească min (m, n). Aceasta deoarece submatricele pătrate ale lui A nu pot în nici un caz să aibă mai multe linii și coloane decît A.

Pentru determinarea practică a rangului unei matrice A, i se aplică acesteia o serie de transformări, a căror proprietate esențială este aceea de a nu modifica rangul.

Asemenea transformări, numite transformări elementare, constau în schimbarea liniilor sau coloanelor între ele, amplificarea sau simplificarea unor linii sau coloane, înlocuirea unei linii (coloane) cu o combinație liniară a altor linii (coloane).

Scanned with OKEN Scanner

Matricei A îi vor fi aplicate succesiv o serie de transformări elementare, urmărindu-se anularea a cît mai multe elemente. Procesul se va opri atunci cînd nu va exista linie sau coloană care să conţină mai mult de un element nenul.

Întrucît rangul rămîne nemodificat în urma aplicării transformărilor susmenţionate, el se va determina pentru ultima matrice obţinută la capătul procesului. De altfel acest lucru se reduce la simpla numărare a elementelor nenule.

Exemplu.

$$A = \begin{Vmatrix} 1 & 5 & -1 & 0 & 1 \\ 2 & 1 & 0 & -1 & 3 \\ 2 & 4 & 1 & 2 & -5 \\ 1 & 8 & 0 & 3 & -7 \end{Vmatrix}.$$

O primă transformare elementară efectuată asupra lui A constă în înlocuirea liniei a patra cu suma liniilor 4, 2 şi a liniilor 1 şi 3, multiplicate cu (-1). Se va obţine o nouă matrice cu acelaşi rang (se va spune echivalentă cu A).

$$A \sim \begin{Vmatrix} 1 & 5 & -1 & 0 & 1 \\ 2 & 1 & 0 & -1 & 3 \\ 2 & 4 & 1 & 2 & -5 \\ 0 & 0 & 0 & 0 & 0 \end{Vmatrix}.$$

În continuare la coloanele 1, 2, 5 se adună coloana 3 amplificată cu 1, 5 respectiv 1,

$$A \sim \begin{Vmatrix} 0 & 0 & -1 & 0 & 0 \\ 2 & 1 & 0 & -1 & 3 \\ 3 & 9 & 1 & 2 & -4 \\ 0 & 0 & 0 & 0 & 0 \end{Vmatrix}.$$

Se adună coloanelor 1, 2, 5 coloana 4 amplificată cu 2, 1 respectiv 3

$$A \sim \begin{Vmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 7 & 11 & 1 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 \end{Vmatrix}.$$

Scanned with OKEN Scanner

Se adună coloanelor 1, 2, 4 coloana 5 amplificată cu $-\frac{7}{2}$, $-\frac{11}{2}$, -1,

$$A \sim \begin{Vmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 \end{Vmatrix}.$$

În sfîrşit, după adunarea liniei 1 la linia 3 se capătă forma

$$A \sim \begin{Vmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 \end{Vmatrix}$$

în care fiecare linie şi coloană conţine cel mult un element nenul. Rangul lui A corespunde numărului elementelor nenule din ultima matrice, adică este 3.

§ 2. REZOLVAREA SISTEMELOR DE ECUAŢII LINIARE

2.1. Forma generală a unui sistem de ecuaţii liniare. Forma matriceală. Forma vectorială.

Un sistem de m ecuaţii liniare cu n necunoscute $x_1, x_2, \ldots, x_n$, se scrie în forma cea mai generală

$$(1) \qquad \begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots & \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m \end{aligned}$$

unde

a_{ij}, $(i = 1, 2, \ldots, m, j = 1, 2, \ldots, n)$ şi b_i $(i = 1, 2, \ldots, m)$ sînt constante cunoscute.

Scanned with OKEN Scanner

A rezolva acest sistem înseamnă a găsi valorile $x_1, x_2, \ldots, x_n$ care să verifice cele m relaţii. După cum sistemul are sau nu soluţie, el se va numi compatibil sau incompatibil.

În prima situaţie interesează de asemenea dacă soluţia este unică sau nu.

Numerele a_{ij}, numite coeficienţii necunoscutelor, se pot ordona într-o matrice, în care pe linii se vor găsi coeficienţii aparţinînd unei aceleiaşi ecuaţii, iar pe coloane, coeficienţii ce înmulţesc o aceiaşi necunoscută.

$$\text{Matricea } A = \left\| \begin{array}{cccc} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{array} \right\| \text{(de ordinul } m \times n)$$

se numeşte *matricea coeficienţilor* sau *matricea sistemului*.

Introducînd în plus notaţiile:

$$X = \left\| \begin{array}{c} x_1 \\ x_2 \\ \vdots \\ x_n \end{array} \right\|$$ reprezentînd matricea coloană $(n \times 1)$ formată din necunoscute, şi

$$b = \left\| \begin{array}{c} b_1 \\ b_2 \\ \vdots \\ b_m \end{array} \right\|$$ pentru matricea-coloană $(m \times 1)$ formată cu termenii liberi ai ecuaţiilor,

sistemul (1) capătă forma matriceală

$$AX = b. \tag{2}$$

Dacă introducem notaţia $P_j = \left\| \begin{array}{c} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{array} \right\|$ pentru coloana j al matricei A, şi ţinem cont de regula de formare a produsului matricelor partiţionate se poate da o nouă formă a sistemului

$$x_1P_1 + x_2P_2 + \ldots + x_nP_n = b. \tag{3}$$

Scanned with OKEN Scanner

2.2. Rezolvarea sistemelor pătrate.

Dacă numărul ecuațiilor coincide cu cel al necunoscutelor, sistemul de ecuații va fi numit pătrat. Pentru acest caz particular, o condiție de compatibilitate și unicitate a soluției este dată de teorema lui Cramer:

Sistemul de ecuații liniare (1) *în care* $m = n$, *este compatibil și are soluție unică dacă și numai dacă* $|A| \neq 0$.

În cazul îndeplinirii acestei condiții, soluția se determină după formulele:

$$x_j = \frac{D_j}{|A|}, \quad j = 1, 2, \ldots, n.$$

unde D_j este determinantul atașat matricei A_j formată din A prin înlocuirea coloanei P_j cu b.

Exemplu.

$$\begin{aligned} 2x_1 - x_2 + x_3 &= 2 \\ x_1 + 2x_2 - x_3 &= 5 \\ -x_1 + 3x_2 + 2x_3 &= -1 \end{aligned}$$

$$A = \begin{Vmatrix} 2 & -1 & 1 \\ 1 & 2 & -1 \\ -1 & 3 & 2 \end{Vmatrix}$$

$$|A| = 8 - 1 + 3 + 2 + 6 + 2 = 20;$$

$$D_1 = \begin{vmatrix} 2 & -1 & 1 \\ 5 & 2 & -1 \\ -1 & 3 & 2 \end{vmatrix} = 8 - 1 + 15 + 2 + 6 + 10 = 40, \quad x_1 = 2$$

$$D_2 = \begin{vmatrix} 2 & 2 & 1 \\ 1 & 5 & -1 \\ -1 & -1 & 2 \end{vmatrix} = 20 + 2 - 1 + 5 - 2 - 4 = 20, \quad x_2 = 1$$

$$D_3 = \begin{vmatrix} 2 & -1 & 2 \\ 1 & 2 & 5 \\ -1 & 3 & -1 \end{vmatrix} = -4 + 5 + 6 + 4 - 30 - 1 = -20, \quad x_3 = -1.$$

Scanned with OKEN Scanner

Folosind forma matriceală (2), dacă $|A| \neq 0$, înseamnă că există inversa A^{-1}. Atunci înmulţind egalitatea

$$AX = b$$

cu A^{-1} (la stînga) se obţine $A^{-1}AX = A^{-1}b$ şi cum $A^{-1}A = E$,

$$X = A^{-1}b$$

relaţie matriceală care dă posibilitatea determinării soluţiei.

În cazul cînd $|A| = 0$, condiţia teoremei lui Cramer nefiind îndeplinită, va rezulta, fie că sistemul este incompatibil fie că are soluţie însă nu este unică.

Răspunsul la această alternativă va fi dat de teorema generală a sistemelor liniare.

2.3. Rezolvarea sistemelor dreptunghiulare

Să presupunem că rangul matricei A este r, $[r \leqslant \min(m,n)]$ şi că există deci o submatrice A_r de ordin $(r \times r)$ a lui A cu $|A_r| \neq 0$, în timp ce toate submatricele A_l cu $l > r$ au determinantul nul.

Determinantul $|A_r|$, (minor al lui A), se va numi *determinant principal* al sistemului.

Vom numi *determinant caracteristic* un determinant de ordin $r + 1$ obţinut prin completarea lui A_r cu o coloană formată din elementele corespunzătoare ale lui b şi cu o linie formată din elementele uneia din liniile lui A care nu au intrat în componenţa submatricei A_r.

Se pot construi deci $m - r$ determinanţi caracteristici. Teorema generală a sistemelor de ecuaţii liniare (a lui Rouché) afirmă:

„Condiţia necesară şi suficientă ca sistemul (1) *să fie compatibil, este ca toţi determinanţii caracteristici să fie nuli. În acest caz soluţia este unică sau nedeterminată după cum* $r = n$ sau $r < n$.

Ecuaţiile corespunzătoare liniilor lui A ce intră în componenţa submatricei A_r sînt denumite *ecuaţii principale*. De asemenea necunoscutele corespunzătoare coloanelor lui A_r se vor zice *necunoscute principale*.

Scanned with OKEN Scanner

În cazul cînd condiţia de compatibilitate a teoremei lui Rouché este îndeplinită, se rezolvă sistemul format din ecuaţiile principale, în raport cu necunoscutele principale.

Dacă $r = n$, toate necunoscutele vor fi principale şi soluţia sistemului va fi unică.

În caz contrar $n - r$ necunoscute vor rămîne arbitrare şi va exista deci o infinitate de soluţii ale sistemului.

Exemplu.

$$\begin{aligned} 2x_1 - x_2 + x_3 &= -1 \\ 3x_1 + x_2 + 2x_3 &= 3 \\ x_1 + 2x_2 + x_3 &= 4 \end{aligned}$$

Avem

$$A = \begin{Vmatrix} 2 & -1 & 1 \\ 3 & 1 & 2 \\ 1 & 2 & 1 \end{Vmatrix}.$$

Rangul lui A este 2. Determinant principal poate fi

$$\begin{vmatrix} 2 & -1 \\ 3 & 1 \end{vmatrix}.$$

Unicul determinant caracteristic este

$$\begin{vmatrix} 2 & -1 & -1 \\ 3 & 1 & 3 \\ 1 & 2 & 4 \end{vmatrix} = 0$$

şi deci conform teoremei lui Rouché sistemul este compatibil.

Primele două ecuaţii şi primele două necunoscute determină sistemul principal

$$\begin{aligned} 2x_1 - x_2 &= -1 - x_3 \\ 3x_1 + x_2 &= 3 - 2x_3 \end{aligned}$$

a cărui soluţie este

$$x_1 = \frac{2 - 3x_3}{5}, \quad x_2 = \frac{9 - x_3}{5}.$$

Scanned with OKEN Scanner

Întrucît x_3 este arbitrară, fiecărei valori atribuite acestei necunoscute îi corespund valorile bine determinate x_1, x_2 toate trei împreună verificînd sistemul.

Deci sistemul este nedeterminat, existînd o infinitate de soluţii.

§ 3. SPAŢII VECTORIALE (reale)

3.1. Vectori. Spaţiul vectorial n-dimensional

Prin vector (real) n-dimensional vom înţelege un sistem ordonat de n numere reale a_i, $(i = 1, 2, \ldots, n)$. În cele ce urmează, referindu-ne la acestea le vom numi componentele vectorului.

Convenim ca vectorii să fie reprezentaţi cu elementele scrise pe linie şi vom utiliza notaţia

$$a = (a_1, a_2, \ldots, a_n).$$

În ceea ce priveşte vectorii coloană de tipul $\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$ sau $\begin{Vmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{Vmatrix}$ îi putem considera obţinuţi prin transpunerea unui vector linie.

Pe parcursul lucrării vom utiliza atît vectori-linie cît şi vectori-coloană, rezultatele prezentate în acest paragraf păstrîndu-se, indiferent de forma reprezentării vectorilor.

Doi vectori sînt egali dacă au acelaşi număr de componente şi dacă, componentele de pe acelaşi loc sînt egale.

Între vectorii cu aceiaşi dimensiune (număr de componente) introducem două operaţii algebrice; adunarea şi înmulţirea cu scalar.

Suma a doi vectori:

$$a = (a_1, a_2, \ldots, a_n) \quad \text{şi} \quad b = (b_1, b_2, \ldots, b_n)$$

va fi vectorul c, ale cărui componente c_i sînt egale cu suma componentelor corespunzătoare ale celor doi vectori. Adică:

$$a + b = (a_1 + b_1, a_2 + b_2, \ldots, a_n + b_n)$$

Scanned with OKEN Scanner

Produsul vectorului $a = (a_1, a_2, \ldots a_n)$ cu scalarul λ, are ca rezultat vectorul

$$\lambda a = (\lambda a_1, \lambda a_2, \ldots, \lambda a_n.)$$

Dacă notăm cu O vectorul nul

$$O = (0, 0, \ldots, 0)$$

putem defini o relaţie de ordine $\geqslant$ prin $a = (a_1, a_2, \ldots, a_n) \geqslant O$, dacă toate componentele a sînt nenegative.

Mulţimea tuturor vectorilor n-dimensionali, înzestrată cu operaţiile algebrice de adunare şi înmulţire cu scalar formează spaţiul vectorial n-dimensional. Va fi desemnat prin simbolul R^n.

3.2. Vectori liniar independenţi. Bază într-un spaţiu vectorial.

Fie un sistem de m vectori n-dimensionali

$$a^1 = (a_1^1, a_2^1, \ldots, a_n^1)$$
$$a^2 = (a_1^2, a_2^2, \ldots, a_n^2)$$
$$\ldots\ldots\ldots\ldots$$
$$a^m = (a_1^m, a_2^m, \ldots, a_n^m)$$

Vectorii $a^1, a^2, \ldots, a^m$ se numesc *liniar-independenţi*, dacă relaţia

$$\lambda_1 a^1 + \lambda_2 a^2 + \ldots + \lambda_m a^m = O \tag{1}$$

este satisfăcută numai pentru $\lambda_1 = \lambda_2 = \ldots = \lambda_m = 0$.

În caz contrar, adică, dacă există m numere reale, $\lambda_1, \lambda_2, \ldots, \lambda_m$ nu toate nule, astfel că

$$\lambda_1 a^1 + \lambda_2 a^2 + \ldots + \lambda_m a^m = O,$$

vectorii se vor numi *liniar dependenţi*.

Să interpretăm acum, relaţia (1), făcînd uz de reprezentarea vectorilor prin componentele lor.

Din definiţia vectorilor şi a înmulţirii cu scalar rezultă că expresia:

$$\lambda_1 a^1 + \lambda_2 a^2 + \ldots + \lambda_m a^m$$

Scanned with OKEN Scanner

este un vector n-dimensional. Egalat cu O, va rezulta anularea tuturor componentelor sale, ceea ce duce la relaţiile

$$\lambda_1 a_1^1 + \lambda_2 a_1^2 + \ldots + \lambda_m a_1^m = 0$$
$$\lambda_2 a_2^1 + \lambda_2 a_2^2 + \ldots + \lambda_m a_2^m = 0$$
$$\ldots\ldots\ldots\ldots\ldots$$
$$\lambda_1 a_n^1 + \lambda_2 a_n^2 + \ldots + \lambda_m a_n^m = 0.$$

Putem interpreta aceste relaţii ca un sistem de n ecuaţii liniare cu m necunoscute

Sistemul fiind omogen, soluţia va fi unică şi anume $\lambda_1 = \lambda_2 = \lambda_3 = \ldots = \lambda_m = 0$, dacă rangul matricei coeficienţilor este m, sau, va admite şi soluţie diferită de cea nulă dacă rangul matricei este inferior lui m.

Cu alte cuvinte, pentru stabilirea independenţei sau dependenţei sistemului de vectori este suficientă calcularea rangului matricei

$$A = \begin{Vmatrix} a_1^1 & a_1^2 & \ldots & a_1^m \\ a_2^1 & a_2^2 & \ldots & a_2^m \\ \ldots & \ldots & \ldots & \ldots \\ a_n^1 & a_n^2 & \ldots & a_n^m \end{Vmatrix},$$

matrice, în care după cum se constată, coloanele sînt formate din componentele vectorilor.

Exemple.

a) Vectorii
$$a^1 = (5, -1, 2, 0),$$
$$a^2 = (0, 1, 3, -1),$$
$$a^3 = (2, 1, 4, 1)$$

sînt liniar independenţi.

Într-adevăr, matricea

$$A = \begin{Vmatrix} 5 & 0 & 2 \\ -1 & 1 & 1 \\ 2 & 3 & 4 \\ 0 & -1 & 1 \end{Vmatrix},$$

Scanned with OKEN Scanner

formată din componentele vectorilor, are rangul trei.

b) Vectorii

$$a^1 = (1, -1, 2),$$
$$a^2 = (0, 3, -2),$$
$$a^3 = (5, -14, 16).$$

sînt liniar dependenţi, rangul matricei

$$A = \begin{Vmatrix} 1 & 0 & 5 \\ -1 & 3 & -14 \\ 2 & -2 & 16 \end{Vmatrix},$$

fiind 2.

Există, bunăoară, numerele 5, -3, -1, astfel ca

$$5a^1 - 3a^2 - 1 \cdot a^3 = O$$

relaţie care se verifică imediat.

Din considerentele anterioare, rezultă că într-un spaţiu vectorial n-dimensional nu pot exista mai mult de n vectori liniar-independenţi.

Aceasta deoarece rangul matricei determinate de orice sistem de m vectori, nu poate depăşi numărul liniilor sale care este n.

Un sistem de n vectori liniar independenţi, dintr-un spaţiu vectorial n-dimensional, R^n, constituie o *bază a* acestui spaţiu.

Cu alte cuvinte o bază este un sistem maximal de vectori liniar independenţi.

De notat, că într-un spaţiu vectorial există o infinitate de baze.

Vectorii

$$e^1 = (1, 0, \ldots, 0),$$
$$e^2 = (0, 1, \ldots, 0),$$
$$\ldots\ldots\ldots\ldots$$
$$e^n = (0, 0, \ldots, 1)$$

formează o bază în R^n. Această afirmaţie este evidentă, deoarece numărul vectorilor din sistem este n, iar rangul matricei

$$\begin{Vmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 1 \end{Vmatrix}$$

Scanned with OKEN Scanner

este n, matricea respectivă fiind chiar matricea unitate de tipul $(n \times n)$.

Întrucît această bază joacă un rol important, în cele ce urmează o vom numi *bază unitate* şi referindu-ne la ea vom nota prescurtat sistemul de vectori $e^1, e^2, \ldots, e^n$, cu E.

Existenţa unei baze într-un spaţiu vectorial, atrage după sine posibilitatea unei anumite reprezentări a vectorilor spaţiului. Această afirmaţie are în vedere următoarea teoremă:

Dacă sistemul de vectori $a^1, a^2, \ldots, a^n$ constituie o bază într-un spaţiu vectorial n-dimensional, atunci orice alt vector a al spaţiului, poate fi reprezentat ca o combinaţie liniară a vectorilor bazei.

$$a = x_1a^1 + x_2a^2 + \ldots + x_na^n. \tag{2}$$

Numerele $x_1, x_2, \ldots, x_n$ se numesc componentele vectorului a în baza $a^1, a^2, \ldots, a^n$.

Sistemul ordonat de numere

$$(x_1, x_2, \ldots, x_n)$$

va reprezenta scrierea vectorului a în baza susmenţionată.

Relaţia vectorială (2) se transcrie pe componente sub forma

$$\begin{aligned} a_1 &= x_1a_1^1 + x_2a_1^2 + \ldots + x_na_1^n \\ a_2 &= x_1a_2^1 + x_2a_2^2 + \ldots + x_na_2^n, \\ &\ldots\ldots\ldots\ldots\ldots\ldots \\ a_n &= x_1a_n^1 + x_2a_n^2 + \ldots + x_na_n^n \end{aligned}$$

care constituie un sistem de n ecuaţii liniare cu n necunoscute $x_1, x_2, \ldots, x_n$.

Deoarece vectorii bazei sînt liniar independenţi, rangul matricei coeficienţilor este egal cu n şi deci sistemul este compatibil, admiţînd o unică soluţie $x_1, x_2, \ldots, x_n$.

Exemplu.

Să se reprezinte vectorul

$$a = (2, -1, 5)$$

în baza

$$\begin{aligned} a^1 &= (1, 1, 0), \\ a^2 &= (2, 0, -1) \\ a^3 &= (3, 1, 2). \end{aligned}$$

Scanned with OKEN Scanner

Vectorii a^1, a^2, a^3 formează o bază în R^3, deoarece matricea

$$\begin{Vmatrix} 1 & 1 & 0 \\ 2 & 0 & -1 \\ 3 & 1 & 2 \end{Vmatrix}$$

are rangul 3.

Pentru determinarea componentelor vectorului a în baza a^1, a^2, a^3, se rezolvă sistemul

$$\begin{aligned} x_1 + 2x_2 + 3x_3 &= 2, \\ x_1 \qquad + x_3 &= -1, \\ - x_2 + 2x_3 &= 5. \end{aligned}$$

Folosind formulele lui Cramer, se deduc uşor

$$x_1 = -\frac{19}{6}, \quad x_2 = \frac{13}{6}, \quad x_3 = \frac{43}{12} \text{ şi deci}$$

$\left(-\frac{19}{6}, \frac{13}{6}, \frac{43}{12}\right)$ este reprezentarea lui a în baza dată.

3.3. Schimbarea coordonatelor unui vector la schimbarea bazei

Întrucît reprezentarea vectorilor este legată de o anumită bază se pune frecvent problema scrierii succesive a aceluiaşi vector în diferite baze.

Fie două baze în R^n date de sistemele de vectori:

$$\begin{aligned} a^1 &= (a_1^1, a_2^1, \ldots, a_n^1) \\ a^2 &= (a_1^2, a_2^2, \ldots, a_n^2), \\ &\ldots\ldots\ldots\ldots \\ a^n &= (a_1^n, a_2^n, \ldots, a_n^n) \end{aligned}$$

şi

$$\begin{aligned} b^1 &= (b_1^1, b_2^1, \ldots, b_n^1) \\ b^2 &= (b_1^2, b_2^2, \ldots, b_n^2), \\ &\ldots\ldots\ldots\ldots \\ b^n &= (b_1^n, b_2^n, \ldots, b_n^n). \end{aligned}$$

Scanned with OKEN Scanner

Fie de asemenea vectorul a a cărui reprezentare în cele două baze este $(x_1, x_2, \ldots, x_n)$ şi respectiv $(y_1, y_2, \ldots, y_n)$.

Stabilind legătura între numerele x_1 şi y_1, găsim regula de schimbare a componentelor unui vector, la schimbarea bazei.

Fie

$$A = \left\| \begin{matrix} a_1^1 & a_1^2 & \ldots & a_1^n \\ a_2^1 & a_2^2 & \ldots & a_2^n \\ \cdot & \cdot & \cdot & \cdot \\ a_n^1 & a_n^2 & \ldots & a_n^n \end{matrix} \right\| \quad \text{şi} \quad B = \left\| \begin{matrix} b_1^1 & b_1 & \ldots & b_1^n \\ b_2^1 & b_2^2 & \ldots & b_2^n \\ \cdot & \cdot & \cdot & \cdot \\ b_n^1 & b_n^2 & \ldots & b_n^n \end{matrix} \right\|$$

matricele ataşate celor două baze.

Deoarece vectorul a se poate scrie ca o combinaţie liniară a vectorilor $a^1, a^2, \ldots, a^n$, sub forma

$$x_1a^1 + x_2a^2 + \ldots + x_na^n$$

şi ca o combinaţie liniară a vectorilor $b^1, b^2, \ldots, b^n$ sub forma

$$y_1b^1 + y_2b^2 + \ldots + y_nb^n$$

rezultă că cele două expresii trebuie să coincidă, ceea ce scris pe componente dă:

$$\begin{aligned} x_1a_1^1 + x_2a_1^2 + \ldots + x_na_1^n &= y_1b_1^1 + y_2b_1^2 + \ldots + y_nb_1^n, \\ x_1a_2^1 + x_2a_2^2 + \ldots + x_na_2^n &= y_1b_2^1 + y_2b_2^2 + \ldots + y_nb_2^n, \\ \cdots & \cdots \\ x_1a_n^1 + x_2a_n^2 + \ldots + x_na_n^n &= y_1b_n^1 + y_2b_n^2 + \ldots + y_nb_n^n. \end{aligned}$$

Cu notaţiile

$$X = (x_1, x_2, \ldots, x_n), \qquad Y = (y_1, y_2, \ldots, y_n)$$

sistemul se scrie

$$A'X = B'Y$$

Cum rangul matricei B este n, rezultă că $|B|$ şi deci şi $|B'|$ este nenul. Ca atare, există matricea inversă $(B')^{-1}$.

Înmulţind relaţia precedentă cu $(B')^{-1}$ (la stînga):

$$\begin{aligned} &(B')^{-1} \cdot A' \cdot X = (B')^{-1} \cdot B' \cdot Y, \text{ adică} \\ &(B')^{-1} \cdot A' \cdot X = Y, \end{aligned}$$

egalitate din care se pot obține componentele y_i ale vectorului a în baza $b^1, b^2, \ldots, b^n$.

O remarcă interesantă este că, dacă $a^1, a^2, \ldots, a^n$ este o bază în R^n, atunci reprezentarea vectorului a^i în această bază este

$$(0, 0, \ldots, 0, 1, 0, \ldots, 0), \text{ 1 aflîndu-se pe locul } i.$$

Exemplu:

a) Fie $(1, -1, 0)$ reprezentarea unui vector a în baza:

$$a^1 = (1, 1, 1),$$
$$a^2 = (1, 1, 0).$$
$$a^3 = (1, 0, 0).$$

Să se scrie vectorul a în baza

$$b^1 = (0, 1, 0),$$
$$b^2 = (3, -1, 2),$$
$$b^3 = (1, 2, 5).$$

$$A' = \begin{Vmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{Vmatrix}, \quad B' = \begin{Vmatrix} 0 & 1 & 0 \\ 3 & -1 & 2 \\ 1 & 2 & 5 \end{Vmatrix};$$

$$B'| = \begin{vmatrix} 0 & 1 & 0 \\ 3 & -1 & 2 \\ 1 & 2 & 5 \end{vmatrix} = -13, (B')^{-1} = \begin{Vmatrix} \frac{9}{13} & \frac{5}{13} & -\frac{2}{13} \\ 1 & 0 & 0 \\ \frac{7}{13} & -\frac{1}{13} & \frac{3}{12} \end{Vmatrix},$$

$$X = \begin{Vmatrix} 1 \\ -1 \\ 0 \end{Vmatrix}.$$

Scanned with OKEN Scanner

Prin urmare

$$Y = (B')^{-1}A' \cdot X = \begin{Vmatrix} \frac{9}{13} & \frac{5}{13} & -\frac{2}{13} \\ 1 & 0 & 0 \\ \frac{7}{13} & -\frac{1}{13} & \frac{3}{13} \end{Vmatrix} \cdot \begin{Vmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{Vmatrix} \cdot \begin{Vmatrix} 1 \\ -1 \\ 0 \end{Vmatrix} =$$

$$= \begin{Vmatrix} -\frac{2}{13} \\ 0 \\ \frac{3}{13} \end{Vmatrix}$$

ceea ce înseamnă că în noua bază b^1, b^2, b^3, vectorul a are forma

$$a = \left(-\frac{2}{13}, \quad 0, \quad \frac{3}{13}\right).$$

3.4. Vectori ortogonali. Matricea de proiecţie.

Fie $a = (a_1, a_2, \ldots, a_n)$ şi $b = (b_1, b_2, \ldots, b_n)$ doi vectori oarecare din spaţiul vectorial n-dimensional R^n.

Prin *produsul scalar* al celor doi vectori înţelegem produsul dintre matricea de ordin $(1 \times n)$ a şi matricea de ordin $(n \times 1)$ b', adică

$$ab' = \sum_{i=1}^{n} a_i b_i.$$

Ţinînd seama de această ultimă expresie a produsului scalar, este uşor de verificat că

$$ab' = ba'.$$

Vectorii a şi b se numesc *ortogonali* dacă produsul lor scalar este nul

$$ab' = ba' = 0.$$

Fie acum r vectori liniar-independenţi $a^1, a^2, \ldots, a^r$ din R^n $(r \leqslant n)$.

Scanned with OKEN Scanner

Notăm

$$D = \left\{ X \in R^n \mid a^i X' = \sum_{j=1}^{n} a_j^i x_j = 0,\ i = 1,\ 2,\ \ldots,\ r \right\}.$$

O ecuație de tipul

$$a^i X' = 0$$

reprezintă un hiperplan H_i (§ 4).

Atunci D ca intersecție a r hiperplane liniar independente este un subspațiu vectorial $(n - r)$-dimensional al lui R^n.

Dacă notăm

$$A_r = \begin{Vmatrix} a^1 \\ a^2 \\ \vdots \\ a^r \end{Vmatrix}$$

atunci D este mulțimea vectorilor X satisfăcînd:

1) $$A_r X' = 0.$$

Fie acum $\tilde{D}$, subspațiul vectorial generat de vectorii $a^1, a^2, \ldots, a^r$ adică

$$\tilde{D} = \left\{ X \in R^n \mid X = \sum_{j=1}^{r} y_j a^j \right\}.$$

Din definiția lui D, rezultă că fiecare vector a^i, $(i = 1,2,\ \ldots,r)$ este ortogonal cu fiecare vector $X \in D$; în consecință orice vector din $\tilde{D}$ va fi ortogonal cu orice vector din D.

Putem spune atunci, că D și D sînt subspații vectoriale ortogonale.

În asemenea condiții orice vector $X \in R^n$ poate fi pus sub forma

2) $$X = X_D + X_{\tilde{D}},\ \text{cu}\ X_D \in D,\ X_{\tilde{D}} \in \tilde{D}.$$

Pentru justificarea egalității 2) să definim matricea

3) $$P_r = E - A_r'(A_r A_r')^{-1} A_r.$$

Scanned with OKEN Scanner

Această definiție are sens, deoarece $A_r A_r'$ este o matrice de ordin $(r \times n)$ nesingulară, deoarece independența vectorilor $a^1, a^2, \ldots, a^r$ implică rang $(A_r) = r$.

Matricea de ordinul $(n \times n)$, dată de egalitatea 3), se numește *matrice de proiecție* pe subspațiul vectorial D.

Să punem acum

$$X_D = XP_r.$$

Se poate verifica imediat că $X_D \in D$.

Într-adevăr

$$A_r X_D' = A_r P' X' \quad A_r E X' - A_r A_r' (A_r A_r')^{-1} A_r X' = A_r X' - A_r X' = 0.$$

Atunci fie

$$4)\ X_{\tilde{D}} = X - X_D = X - XP_r = X(E - P_r) = XA_r'(A_r A_r')^{-1} A_r.$$

Matricea $P_r = A_r'(A_r A_r')^{-1} A_r$ se numește matrice de proiecție pe subspațiul vectorial $\tilde{D}$.

Se poate verifica imediat că $X_{\tilde{D}} \in \tilde{D}$

Într-adevăr: $XA_r'(A_r A_r')^{-1}$ este un vector linie r-dimensional. Ca atare, egalitatea

$$X_{\tilde{D}} = [X - A'(A_r A')^{-1}] A_r.$$

nu arată altceva, decît că $X_{\tilde{D}}$ poate fi reprezentat ca o combinație liniară a vectorilor $a^1, a^2, \ldots, a^r$

În concluzie, deoarece $X_D \in D$ și $X_{\tilde{D}} \in \tilde{D}$, prima egalitate din 4) arată că reprezentarea oricărui vector X sub forma dată de 2), este posibilă.

§ 4. MULȚIMI ȘI FUNCȚII CONVEXE

O mulțime C de vectori din spațiul R^n este numită *convexă*, dacă pentru oricare doi vectori X^1, X^2 din C și pentru orice număr real $0 \leqslant \lambda \leqslant 1$, vectorul $X = \lambda X^1 + (1 - \lambda) X^2$ aparține de asemenea lui C.

Scanned with OKEN Scanner

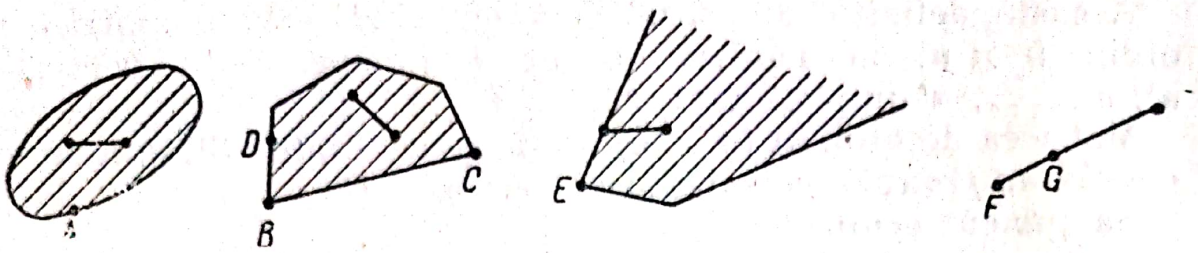


Fig. 1

Obişnuit, referindu-ne la vectorii aparţinînd unei mulţimi, îi vom numi şi puncte ale mulţimii.

Mulţimea vectorilor X de forma $\lambda X^1 + (1 - \lambda)X^2$, pentru X^1 şi X^2 daţi şi λ variabil între 0 şi 1, constituie segmentul determinat de punctele X^1 şi X^2.

Definiţia unei mulţimi convexe se poate da atunci şi sub forma:

C este *convexă* dacă odată cu oricare două puncte ale sale, conţine întregul segment determinat de ele.

Pentru a da o imagine intuitivă a acestei noţiuni să considerăm spaţiul vectorial R^2.

Întrucît oricărui sistem de două numere reale (x, y) îi corespunde un punct în plan, reprezentarea geometrică a spaţiului vectorial bi-dimensional este planul raportat la un sistem de coordonate.

Este uşor de verificat că punctele din plan, situate pe segmentul determinat de două puncte date de coordonate (x_1, y_1) respectiv (x_2, y_2), au coordonatele de forma:

$$x = \lambda x_1 + (1 - \lambda)x_2,$$
$$y = \lambda y_1 + (1 - \lambda)y_2, \quad \text{cu } 0 \leqslant \lambda \leqslant 1,$$

relaţii care se scriu vectorial

$$(x, y) = \lambda(x_1, y_1) + (1 - \lambda)(x_2, y_2).$$

Figurile 1 şi 2 reprezintă mulţimi convexe şi respectiv neconvexe din plan.

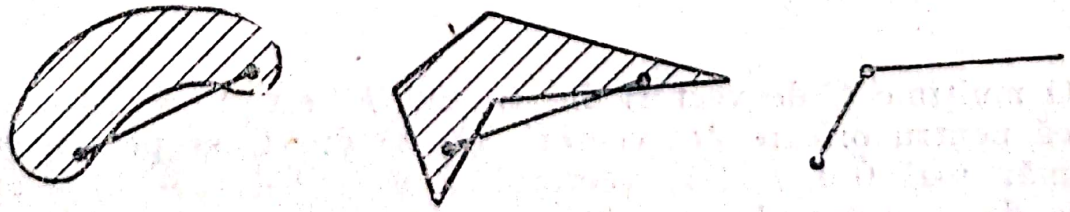


Fig. 2

Scanned with OKEN Scanner

Vom numi *combinație liniară convexă* a vectorilor $X^1, X^2, \ldots, X^m$ orice expresie de forma

$$\lambda_1 X^1 + \lambda_2 X^2 + \ldots + \lambda_m X^m, \text{ cu } \lambda_i \geqslant 0,\ i = 1, 2, \ldots, m$$

$$\text{și } \sum_{i=1}^{m} \lambda_i = 1.$$

Din definiția mulțimii convexe rezultă că dacă $X^1, X^2, \ldots, X^m$, aparțin lui C, atunci orice combinație liniară convexă, a acestor vectori este de asemenea un punct al lui C.

Între proprietățile mulțimilor convexe, o notăm și pe aceea potrivit căreia, intersecția a două sau mai multe mulțimi convexe este tot o mulțime convexă.

Se numește *punct frontieră* a unei mulțimi convexe C, un vector X care are proprietatea că orice vecinătate a sa conține atît puncte din C cît și puncte ce nu aparțin lui C.

Un punct X al mulțimii convexe C este *punct extrem* sau *vîrf*, dacă nu există X^1 și X^2 din C astfel ca

$$X = \lambda X^1 + (1 - \lambda) X^2, \quad \text{cu } 0 < \lambda < 1.$$

În figura 1, punctele A, D, G sînt puncte frontieră, iar punctele B, C, E, F vîrfuri ale mulțimilor convexe respective. Mulțimea vectorilor $X = (x_1, x_2, \ldots, x_n)$ din R^n care satisfac o relație de tipul

$$a_1 x_1 + a_2 x_2 + \ldots + a_n x_n = \lambda$$

formează un *hiperplan*.

Cu notația $a = (a_1, a_2, \ldots, a_n)$, ecuația unui hiperplan este

$$aX' = \lambda$$

În cazul planului, hiperplanul

$$a_1 x + a_2 y = \lambda$$

reprezintă o dreaptă.

Mulțimea vectorilor X care verifică relația:

$$aX' \leqslant \lambda \qquad \text{sau } aX' \geqslant \lambda$$

formează un *semispațiu*.

Scanned with OKEN Scanner

În R^2, semispațiile sînt semiplanele definite de inecuațiile $a_1x + a_2y \leqslant \lambda$ respectiv $a_1x + a_2y \geqslant \lambda$.

Intersecția a m hiperplane, adică mulțimea vectorilor X din R^n, satisfăcînd sistemul de relații

$$(1) \qquad \begin{aligned} a^1X^1 &\leqslant \lambda_1, \\ a^2X^2 &\leqslant \lambda_2, \\ &\cdots\cdots \\ a^mX^m &\leqslant \lambda_m \end{aligned}$$

se numește *tronson convex*.

Relațiile (1) pot fi scrise și sub forma

$$AX' \leqslant \lambda$$

unde

$$A = \left\| \begin{matrix} a^1 \\ a^2 \\ \vdots \\ a^m \end{matrix} \right\| \text{ este o matrice de tip } (n \times n)$$

ar

$$\lambda = \left\| \begin{matrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_m \end{matrix} \right\|.$$

Este ușor de verificat că hiperplanul, semispațiul și tronsonul sînt mulțimi convexe.

Din definiția tronsonului și a punctului extrem, rezultă că un vîrf al unui tronson este un vector X, ale cărei componente satisfac cu egalitatea n din cele m inecuații din (1). Adică un vîrf al unui tronson se găsește la intersecția a n din cele m hiperplane ce mărginesc tronsonul.

Un tronson mărginit se numește *poliedru convex*. O proprietate importantă a poliedrului convex este aceea că numărul vîrfurilor sale este finit.

Scanned with OKEN Scanner

De asemenea se demonstrează că orice punct al unui poliedru convex se poate reprezenta ca o combinaţie liniară convexă a vîrfurilor sale.

Fie f o funcţie definită pe domeniul convex C cu valori reale. este o *funcţie convexă*, dacă oricare ar fi punctele X_1, X_2 din C şi numărul $\lambda \in [0, 1]$, este verificată inegalitatea

$$f[\lambda X_1 + (1 - \lambda)X_2] \leqslant \lambda f(X_1) + (1 - \lambda)f(X_2). \tag{2}$$

Această proprietate definitorie a funcţiilor convexe, poate fi uşor pusă sub forma, mai generală

$$f\left(\sum_{i=1}^{m} \lambda_i X_i\right) \leqslant \sum_{i=1}^{m} \lambda_i f(X_i) \tag{2'}$$

oricare ar fi $X_1, X_2, \ldots, X_m \in C$, $\lambda_1, \lambda_2, \ldots, \lambda_m \in [0, 1]$ cu $\sum_{i=1}^{m} \lambda_i = 1$.

O funcţie definită pe C este *concavă*, dacă $-f$ este convexă, adică oricare ar fi punctele $X_1, X_2, \in C$ şi $\lambda \in [0, 1]$, este verificată inegalitatea

$$f[\lambda X_1 + (1 - \lambda)X_2] \geqslant \lambda f(X_1) + (1 - \lambda)f(X_2). \tag{3}$$

O funcţie f este *strict convexă* (strict concavă) dacă inegalitatea (2), (3) este verificată întotdeauna fără egalitate.

Dintre proprietăţile funcţiilor convexe citäm:

— Orice mulţime de tipul $\{X \in C | f(X) \leqslant \alpha, \alpha \in R\}$ este o mulţime convexă.

— Orice minim local al unei funcţii convexe definită pe un domeniu convex C, este un minim global.

— O funcţie strict convexă are un singur minim local.

— Dacă funcţia convexă $f(X)$ admite derivate parţiale în raport cu fiecare componentă a argumentului său, şi dacă $X^1 = (x_1^1, x_2^1, \ldots, x_n^1)$, $X^2 = (x_1^2, x_2^2, \ldots, x_n^2)$ sînt două puncte ale domeniului convex C, atunci

$$f(X^1) - f(X^2) \geqslant \sum_{i=1}^{n} (x_i^1 - x_i^2) \frac{\partial f}{\partial x_i}(X^1).$$

Scanned with OKEN Scanner

În forma sa cea mai generală un sistem de m inecuații liniare cu n necunoscute $x_1, x_2, \ldots, x_n$ se scrie

$$\begin{aligned}
&a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leqslant b_1,\\
&a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leqslant b_2,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&a_{l1}x_1 + a_{l2}x_2 + \ldots + a_{ln}x_n \leqslant b_l,\\
&a_{l+1,1}x_1 + a_{l+1,2}x_2 + \ldots + a_{l+1,n}x_n \geqslant b_{l+1},\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \geqslant b_m.
\end{aligned}$$

Sistemul este compatibil dacă există n numere reale $x_1, x_2, \ldots x_n$ care să satisfacă simultan cele m relații.

Un sistem de inecuații liniare poate fi adus la una din formele

$$\begin{aligned}
&a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leqslant b_1,\\
&a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leqslant b_2,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \leqslant b_m
\end{aligned} \tag{1}$$

sau

$$\begin{aligned}
&a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \geqslant b_1,\\
&a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \geqslant b_2,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \geqslant b_m
\end{aligned} \tag{2}$$

în care toate inecuațiile sînt de același sens. Transformarea se face amplificînd cu (-1) inecuațiile de sens contrar.

De asemenea (1) poate fi adus la forma (2) sau reciproc.

Cu aceste observații, determinarea soluțiilor unui sistem de inecuații de forma (1), rezolvă complet problema sistemelor de inecuații liniare.

Scanned with OKEN Scanner

Cu notaţiile

$$A = \left\| \begin{matrix} a_{11}a_{12} & \dots & a_{1n} \\ a_{21}a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots \\ a_{m1}a_{m2} & \dots & a_{mn} \end{matrix} \right\|, \quad X = \left\| \begin{matrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{matrix} \right\|, \quad b = \left\| \begin{matrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{matrix} \right\|$$

(1) se scrie $AX \leqslant b$, ceea ce arată că mulţimea vectorilor X care satisfac această relaţie este un tronson convex.
Deci:

Mulţimea soluţiilor unui sistem de inecuaţii liniare este convexă.

Pentru rezolvarea sistemului (1) se consideră sistemul ajutător de ecuaţii liniare

$$\begin{aligned} &a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + y_1 = b_1, \\ &a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n + y_2 = b_2, \\ &\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ &a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n + y_m = b_m. \end{aligned} \tag{1'}$$

Sau sub formă matriceală:

$$\overline{A}Z = b$$

unde $\overline{A} = ||AE_m||$ iar $Z = \left\| \begin{matrix} X' \\ y_1 \\ y_2 \\ \vdots \\ y_m \end{matrix} \right\|$.

Deoarece din (1),

$$b_i - (a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n) \geqslant 0, \quad i = 1,2,\dots,m$$

cantităţile $y_1, y_2, \dots, y_m$, care se adaugă la membrul stîng al inecuaţiilor în scopul stabilirii egalităţii, se vor presupune pozitive. (1') va constitui deci, un sistem de m ecuaţii liniare cu $n + m$ necunoscute $x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m$, ultimele m fiind obligatorii nenegative. Vom numi necunoscutele $y_1, y_2, \dots, y_m$ variabile *auxiliare* (de compensaţie).

Scanned with OKEN Scanner

Teorema 1. *Dacă* $\overline{X} = \begin{Vmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_m \end{Vmatrix}$ *este o soluție a sistemului de ine-*

cuații (1), *atunci există o soluție a sistemului de ecuații liniare* (1′) *de forma*

$$\overline{Z} = \begin{Vmatrix} \overline{X} \\ \bar{y}_1 \\ \bar{y}_2 \\ \vdots \\ \bar{y}_m \end{Vmatrix} \text{ cu } \bar{y}_i \geqslant 0 \quad (i = 1, 2, \ldots, m).$$

Demonstrație. Deoarece $\overline{X}$ este soluție a sistemului (1) rezultă:

$$A\overline{X} \leqslant b,$$

adică

$$b - A\overline{X} \geqslant 0.$$

Înlocuind în (1′) pe $x_1, x_2, \ldots, x_n$ cu $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n$ și ținînd cont de regula de înmulțire a matricelor partiționate, rezultă

$$A\overline{X} + E_m \cdot \begin{Vmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{Vmatrix} = b$$

sau

$$A\overline{X} + \begin{Vmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{Vmatrix} = b, \text{ adică } \begin{Vmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{Vmatrix} = b - A\overline{X}$$

și cum matricea din membrul drept este $\geqslant 0$, rezultă că vectorul $\begin{Vmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{Vmatrix}$ are componente nenegative.

Scanned with OKEN Scanner

Teorema 2. *Dacă* $\overline{Z} = \begin{Vmatrix} \overline{X} \\ y_1 \\ y_2 \\ \vdots \\ y_m \end{Vmatrix}$ *este o soluție a sistemului* (1′) *în care* $y_1 \geqslant 0,\ y_2 \geqslant 0,\ \ldots,\ y_m \geqslant 0,$ *atunci* $\overline{X} = \begin{Vmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_m \end{Vmatrix}$ *este o soluție a sistemului de inecuații* (1).

Demonstrație. Dacă $\overline{Z}$ este soluție a sistemului ajutător (1′) din relațiile precedente rezultă că

$$b - A\overline{X} = \begin{Vmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{Vmatrix}$$

și cum matricea coloană din membrul drept are elemente nenegative, rezultă

$$b - A\overline{X} \geqslant 0 \quad \text{sau} \quad A\overline{X} \leqslant b.$$

Aceste două teoreme, ne dau posibilitatea de a rezolva un sistem de inecuații liniare, deducînd soluția acestuia din cea a sistemului de ecuații liniare ajutător.

Exemplu:

$$\begin{aligned} 2x_1 + x_2 - x_3 &\leqslant 5, \\ x_1 - x_2 + 3x_3 &\leqslant 6, \\ -x_1 + 3x_2 + 2x_3 &\leqslant 3. \end{aligned}$$

Matricea sistemului ajutător

$$\begin{aligned} 2x_1 + x_2 - x_3 + y_1 &= 5, \\ x_1 - x_2 + 3x_3 + y_2 &= 6, \\ -x_1 + 3x_2 + 2x_3 + y_3 &= 3, \end{aligned}$$

Scanned with OKEN Scanner

este

$$A = \left\| \begin{array}{cccccc} 2 & 1 & -1 & 1 & 0 & 0 \\ 1 & -1 & 3 & 0 & 1 & 0 \\ -1 & 3 & 2 & 0 & 0 & 1 \end{array} \right\|$$

care are evident rangul 3 (deoarece conține matricea unitate).

Se poate lua ca determinant principal

$$\begin{vmatrix} 2 & 1 & -1 \\ 1 & -1 & 3 \\ -1 & 3 & 2 \end{vmatrix} = -29$$

ceea ce duce la rezolvarea sistemului principal.

$$\begin{aligned} 2x_1 + x_2 - x_3 &= 5 - y_1, \\ x_1 - x_2 + 3x_3 &= 6 - y_2, \\ -x_1 + 3x_2 + 2x_3 &= 3 - y_3. \end{aligned}$$

Soluția va fi

$$x_1 = \frac{79 - 11y_1 - 5y_2 + 2y_3}{29}, \quad x_2 = \frac{28 - 5y_1 + 3y_2 - 7y_3}{29}, \quad x_3 = \frac{41 + 2y_1 - 7y_2 - 3y_3}{29}.$$

y_1, y_2, y_3 fiind arbitrare dar pozitive.

§ 6. FORME LINIARE ȘI PĂTRATICE

6.1. Forme liniare

O funcție reală de n variabile este o lege care asociază oricărui vector $X = (x_1, x_2, \ldots, x_n)$ din R^n un număr real bine determinat, notat simbolic $f(x_1, x_2, \ldots, x_n)$ sau $f(X)$.

Mulțimea tuturor vectorilor X pe care este definită funcția se va numi domeniul de definiție al funcției f.

Fie funcția f definită pe domeniul $D \subset R^n$.

Scanned with OKEN Scanner

Un vector $X^* \in D$ este denumit *punct de maxim* (absolut) al lui f pe D, dacă $f(X) \leqslant f(X^*)$ pentru orice vector X din D. $f(X^*)$ se va numi maximul lui f.

Analog, vectorul $X^* \in D$ este *punct de minim* (absolut) al lui f, pe D, dacă

$$f(X) \geqslant f(X^*) \text{ pentru orice } X \in D.$$

$f(X^*)$ se va numi *minimul* lui f.

O funcţie de n variabile $f(X)$, se numeşte formă liniară, dacă

$$f(\lambda X) = \lambda f(X) \text{ şi } f(X + Y) = f(X) + f(Y)$$

oricare ar fi numărul real λ şi vectorii X şi Y din D.

În problemele de programare liniară apar funcţii de n variabile de forma

$$f(x_1, x_2, \ldots, x_n) = c_1x_1 + c_2x_2 + \ldots + c_nx_n$$

sau vectorial

$$f(X) = CX \text{ unde } C = (c_1, c_2, \ldots, c_n)$$

(X este considerat vector-coloană).

Teorema 1. *Funcţia $f(X) = CX$ este o formă liniară.*

Demonstraţie. Fie λ real. Atunci

$$f(\lambda X) = C(\lambda X) = C\lambda X = \lambda CX = \lambda f(X).$$

De asemenea dacă $X, Y \in R^n$

$$f(X + Y) = C(X + Y) = CX + CY = f(X) + f(Y).$$

Fără demonstraţie dăm şi următoarea teoremă

Teorema 2. *Forma liniară $f(X) = CX$, definită pe o mulţime convexă D, îşi atinge maximul şi minimul pe frontiera lui D.*

Demonstraţia va fi dată într-un caz particular în capitolul II.

6.2. Forme pătratice.

Numim *formă pătratică*, o funcţie de n variabile reale de forma

$$Q(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}x_ix_j.$$

Scanned with OKEN Scanner

Cu notaţiile matriceale

$$C = \begin{Vmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots \\ c_{n1} & c_{n2} & \dots & c_{nn} \end{Vmatrix}, \quad X = \begin{Vmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Vmatrix}$$

putem scrie

$$Q(X) = X'CX.$$

Vedem astfel, că fiecărei forme pătratice i se asociază o matrice pătrată.

O matrice pătrată C se numeşte *nenegativ definită* (sau semipozitiv definită) dacă oricare ar fi $x \in R^n$,

$$X'CX \geqslant 0. \tag{1}$$

O matrice pătrată C se numeşte *pozitiv definită* (sau strict pozitiv definită) dacă în relaţia (1) are loc inegalitatea strictă.

O matrice C este *negativ semi-definită* dacă $-C$ este pozitiv semi-definită.

Se poate arăta că dacă C este o matrice pozitiv definită, atunci toţi determinanţii principali ai săi sînt pozitivi, adică:

$$|C_1| = c_{11} > 0;$$

$$|C_2| = \begin{vmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{vmatrix} > 0; \quad |C_3| = \begin{vmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{vmatrix} > 0.$$

$$\dots \; |C_n| = |C| > 0.$$

Rezultă de aici:

Orice matrice pozitiv definită este *nesingulară*.

Forma pătratică $Q(X) = X'CX$ se va numi *nenegativ* (*pozitiv*) *definită*, dacă matricea C este nenegativ (pozitiv) definită.

Un rezultat important utilizat în această lucrare (cap. III). este următorul:

Teorema 1. *Fie forma pătratică*

$$Q(X) = X'CX.$$

Dacă matricea C este nenegativ definită, atunci $Q(X)$ este o funcţie convexă. Dacă C este pozitiv definită atunci $Q(X)$ este strict convexă.

Scanned with OKEN Scanner

Demonstrație. Fie $X^1,\ X^2 \in R^n$ și $0 \leqslant \lambda \leqslant 1$.
Atunci

$$\lambda Q(X^1) + (1-\lambda)Q(X^2) = \lambda(X^1)'CX^1 + (1-\lambda)(X^2)'CX^2 =$$
$$= \lambda(X^2)'C(X^1 - X^2) + \lambda(X^1 - X^2)'CX^2 + \lambda(X^1 - X^2)'C(X^1 -$$
$$- X^2) + (X^2)'CX^2 \geqslant \lambda(X^2)'C(X^1 - X^2) + \lambda(X^1 - X^2)'CX^2 +$$
$$+ \lambda^2(X^1 - X^2)'C(X^1 - X^2) + X^2CX^2 = [\lambda(X^1 - X^2) +$$
$$+ X^2]'C[\lambda(X^1 - X^2) + X^2] = [\lambda X^1 + (1-\lambda)X^2]'C[\lambda X^1 +$$
$$+ (1-\lambda)X^2] = Q[\lambda X^1 + (1-\lambda)X^2].$$

§.7. ELEMENTE DE TEORIA PROBABILITĂȚILOR

Vom reprezenta un cîmp de probabilitate prin tripletul $\{\Omega, \mathfrak{K}, P\}$, unde Ω este evenimentul sigur, $\mathfrak{K}$ este un corp borelian de părți ale lui Ω, iar P o probabilitate definită pe $\{\Omega, \mathfrak{K}\}$.

În teoria axiomatică a probabilităților, dezvoltată pe baza definițiilor date de Kolmogorov, Ω este asimilat cu o mulțime ale cărei puncte ω le vom numi *evenimente elementare*.

O *variabilă aleatoare* unidimensională, apare în acest context ca o funcție definită pe Ω cu valori reale, măsurabilă în raport cu spațiul măsurabil $\{\Omega, \mathfrak{K}\}$.

Analog, o variabilă aleatoare n-dimensională este o funcție definită pe Ω cu valori în R^n. Se poate scrie atunci

$$f = (f_1, f_2, \ldots, f_n)$$

cu f_i variabile aleatoare unidimensionale $(i = 1, 2, \ldots, n)$.

Funcția de repartiție a variabilei aleatoare f, este funcția: $F : R \to R$ definită prin

$$F(x) = P(f < x), \text{ oricare ar fi } x \in R.$$

Analog, funcția de repartiție a variabilei aleatoare n-dimensionale $f = (f_1, f_2, \ldots, f_n)$ se definește prin

$$F(x_1, x_2, \ldots, x_n) = P(f_1 < x_1, f_2 < x_2, \ldots, f_n < x_n).$$

Densitatea de repartiție a variabilei aleatoare f este funcția $f(x)$ definită de egalitatea

$$F(x) = \int_{-\infty}^{x} \rho(t)\,dt.$$

Scanned with OKEN Scanner

Legea de probabilitate a unei variabile aleatoare este descrisă, matematic, de funcţia sa de repartiţie, sau de densitatea de repartiţie (dacă există).

Dacă mulţimea valorilor variabilei aleatoare f este finită sau numărabilă, atunci f este o variabilă aleatoare de tip discret.

Legea de probabilitate a ei se exprimă de obicei sub forma tabloului de repartiţie

$$f\begin{pmatrix} x_1, & x_2, & \ldots, & x_n, & \ldots \\ p_1, & p_2, & \ldots, & p_n, & \ldots \end{pmatrix}$$

unde $p_k = P(f = x_k)$.

Dacă mulţimea valorilor lui f este nenumărabilă, atunci f este o variabilă aleatoare de tip continuu.

Exemple de legi de probabilitate de tip continuu:

— Legea uniformă pe intervalul $[a, b]$ se defineşte prin densitatea de repartiţie a variabilei, de forma

$$\rho(x) = \begin{cases} \dfrac{1}{b-a} \text{ pentru } x \in [a, b], \\ 0 \text{ pentru } x \notin [a, b]. \end{cases}$$

— Legea normală de parametrii m, σ, $(N(m, \sigma))$ este caracterizată de densitatea de repartiţie

$$\rho(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(x-m)^2}{2\sigma^2}}.$$

Valoarea medie a variabilei aleatoare f este numărul dat de expresia

$$M(f) = \int_{-\infty}^{+\infty} x\,dF(x) = \int_{-\infty}^{+\infty} x\rho(x)dx.$$

În cazul repartiţiilor discrete, integrala se înlocuieşte cu o sumă

$$M(f) = \sum_k x_k p_k.$$

Scanned with OKEN Scanner

Capitolul II

PROGRAMAREA LINIARĂ

§ 1. FORMULAREA PROBLEMEI. EXEMPLE

Dintre problemele programării matematice, de un deosebit interes se bucură aceea cunoscută sub numele de *programare liniară*.

O problemă de programare liniară, încadrîndu-se în limitele generale ale modelelor problemelor programării matematice, se individualizează în mulţimea acestora datorită faptului că atît restricţiile, cît şi funcţia de eficienţă, se exprimă matematic sub forma unor funcţii liniare.

Această formă specifică a expresiei sale matematice, a făcut posibilă folosirea în acest caz a unui aparat matematic destul de simplu, furnizat de algebra liniară, şi elaborarea unor metode eficiente de determinare a soluţiilor problemei.

Algoritmul simplex, prima metodă de rezolvare a problemei, datorat lui *G. Dantzig*, face posibilă aflarea soluţiilor problemei de programare liniară, în cazul cînd acestea există, sau poate dovedi neexistenţa soluţiilor, degenerarea lor etc.

Ulterior, metodele de rezolvare s-au diversificat, unele din ele căutînd să răspundă condiţiilor impuse de introducerea tehnicii de calcul la maşinile electronice.

Dar, interesul suscitat de problema programării liniare nu este datorat numai facilităţilor de ordin matematic care apar în studiul ei, ci şi faptului că o gamă largă de fenomene din ştiinţele economice şi sociale, conduc, după modelarea lor matematică, la acest tip de problemă.

Să urmărim în continuare cîteva exemple, bineînţeles simplificate, dar deloc banale, de probleme economice concrete, des întîlnite în practică, a căror rezolvare corectă se poate face numai prin aplicarea metodelor matematice ale programării liniare.

Scanned with OKEN Scanner

Exemplul I (Problema nutriţiei)

Din considerente biologice s-a stabilit că raţia zilnică, a fiecărui animal crescut de către o fermă zootehnică, trebuie să conţină n elemente nutritive $N_1, N_2, \ldots, N_n$ în cantităţile $b_1, b_2, \ldots, b_n$.

Ferma dispune de m tipuri de nutreţuri $A_1, A_2, \ldots, A_m$, pe care le procură la preţurile $c_1, c_2, \ldots, c_m$.

Analiza conţinutului în substanţe nutritive arată că o unitate din nutreţul A_j, $(j = 1, 2, \ldots, m)$ conţine a_{ji} unităţi de substanţă de tipul i $(i = 1, 2, \ldots, n)$.

Se pune problema hrănirii raţionale a animalelor, adică alcătuirea unor raţii care să corespundă cerinţelor biologice şi în acelaşi timp, să implice cheltuieli minime.

Pentru modelarea matematică a problemei, să notăm cu $x_1, x_2, \ldots, x_m$ cantităţile de nutreţ din tipurile $A_1, A_2, \ldots, A_m$ ce se vor aloca pentru hrana zilnică a unui animal.

Atunci necesităţile biologice enunţate, vor impune următoarele restricţii asupra necunoscutelor $x_1, x_2, \ldots, x_m$.

$$\begin{aligned} a_{11}x_1 + a_{21}x_2 + \ldots + a_{m1}x_m &\geqslant b_1. \\ a_{12}x_1 + a_{22}x_2 + \ldots + a_{m2}x_m &\geqslant b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{1n}x_1 + a_{2n}x_2 + \ldots + a_{mn}x_m &\geqslant b_n. \end{aligned}$$

Într-adevăr, deoarece substanţa nutritivă N_i, $(i = 1, 2, \ldots, n)$ se găseşte în fiecare nutreţ $A_j (j = 1, 2, \ldots, m)$ în cantitatea a_{ji} unităţi la unitatea de nutreţ, atunci în întreaga raţie, se va găsi într-o cantitate totală egală cu $\sum_{j=1}^{m} a_{ji}x_j$. Or această cantitate trebuia să fie cel puţin egală cu minimul necesar b_i, pentru a fi respectate normele biologice prescrise.

Acest raţionament, repetat pentru fiecare din cele n substanţe nutritive luate în considerare, ne conduce la sistemul de condiţii (1).

În acelaşi timp, prin însăşi notaţiile introduse, este evident că

$$(2) \qquad x_j \geqslant 0, \qquad j = 1, 2, \ldots, m.$$

Considerentele economice presupun minimizarea costului unei raţii, cost care va fi dat de expresia

$$z = c_1x_1 + c_2x_2 + \ldots + c_mx_m.$$

Aceasta deoarece costul unei unităţi din nutreţul A_j este c_j şi ca atare, costul întregii cantităţii din acest nutreţ care participă la alcătuirea raţiei va fi $c_j x_j$.

În concluzie, aspectul matematic al problemei puse iniţial va fi următorul:

— să se determine valorile necunoscutelor $x_1, x_2, \ldots, x_m$, care satisfac condiţiile (1), (2), şi:

$$\text{(3)} \qquad \text{minim}\left(z = \sum_{j=1}^{m} c_j x_j\right).$$

O analiză sumară a problemei scoate în evidenţă faptul că, dacă nu conţin elemente contradictorii, condiţiile 1) şi 2) pot fi satisfăcute pentru o infinitate de seturi de valori ale celor m variabile. Aceasta, deoarece, din punct de vedere matematic, cele două grupuri de condiţii reprezintă un sistem de inegalităţi liniare. Or, este de aşteptat ca pentru valori diferite atribuite variabilelor $x_1, x_2, \ldots, x_m$ şi valorile lui z să fie diferite.

Drept urmare, aflarea soluţiei optime a problemei, adică determinarea acelor valori ale variabilelor pentru care z să capete valoarea minimă, devine o problemă ce nu se poate rezolva pe cale intuitivă.

Exemplul II (*Problema amestecului*)

Pentru ilustrarea acestei probleme practice, să recurgem la date numerice fictive.

O intreprindere de rafinare a petrolului produce zilnic patru tipuri de rafinate din petrol B_1, B_2, B_3, B_4, care servesc drept materii prime pentru procesele ulterioare de producţie.

Capacităţile zilnice de producţie şi preţurile de producţie ale celor patru sortimente sînt date în tabloul următor:

materia primă	*capacitatea de producţie zilnică (în tone)*	*preţul de producţie (în lei/t)*
B_1	200	210
B_2	280	260
B_3	250	240
B_4	300	180

Scanned with OKEN Scanner

În final, rafinăria livrează două tipuri de benzine obținute prin combinarea celor patru tipuri de produse inițiale, după următoarea rețetă:

Pentru benzina A se folosește un amestec în care B_2 se găsește în proporție de cel puțin 40%, B_3 cel puțin 35%, iar B_4 cel puțin 15%.

Pentru benzina B se combină B_1 în proporție de cel puțin 15% cu B_2 cel mult 45% și B_4 cel mult 25%.

Produsele finite se vînd la prețul de 270, respectiv 320 lei tona.

Se cere planificarea cantităților de benzine de tipurile A și B ce trebuie produse zilnic, pentru ca venitul net al întreprinderii să fie maxim.

Pentru modelarea problemei, notăm:

x_1 = cantitatea de produs B_1 folosită pentru prepararea benzinei B (în tone);

x_2 = cantitatea de produs B_2 folosită pentru prepararea benzinei A;

x_3 = cantitatea de produs B_2 folosită pentru prepararea benzinei B;

x_4 = cantitatea de produs B_3 folosită pentru prepararea benzinei A;

x_5 = cantitatea de produs B_4 folosită pentru prepararea benzinei A;

x_6 = cantitatea de produs B_4 folosită pentru prepararea benzinei B.

Atunci putem scrie restricțiile problemei, adică condițiile la care trebuie să răspundă cele șase variabile, condiții deduse din însăși formularea problemei concrete.

Restricțiile impuse de limitarea capacităților de producție sînt:

$$(1_a)\qquad \begin{aligned} x_1 &\leqslant 200,\\ x_2 + x_3 &\leqslant 280,\\ x_4 &\leqslant 250,\\ x_5 + x_6 &\leqslant 300. \end{aligned}$$

Membrul stîng al fiecărei inecuații din acest grup specifică cantitatea totală din fiecare tip de produs B_1, B_2, B_3, B_4

Scanned with OKEN Scanner

folosită la elaborarea benzinelor A și B, cantitate ce trebuie să nu depășească capacitatea zilnică de producție.

Alt grup de restricții provine din rețeta de fabricație.

Întreaga cantitate de benzină de tipul A planificată este egală cu $x_2+x_4+x_5$ (deoarece amestecul se face fără schimbări cantitative), iar cantitatea totală de benzină B va fi egală cu $x_1 + x_3 + x_5$. Transformînd procentele în fracții ordinare, obținem:

$$\left.\begin{aligned} x_2 &\geqslant \frac{40}{100}(x_2 + x_4 + x_5) \\ x_4 &\geqslant \frac{35}{100}(x_2 + x_4 + x_5) \\ x_5 &\geqslant \frac{15}{100}(x_2 + x_4 + x_5) \end{aligned}\right\} \text{rețeta } A.$$

$$\left.\begin{aligned} x_1 &\geqslant \frac{15}{100}(x_1 + x_3 + x_6) \\ x_3 &\leqslant \frac{45}{100}(x_1 + x_3 + x_6) \\ x_6 &\leqslant \frac{25}{100}(x_1 + x_3 + x_6) \end{aligned}\right\} \text{rețeta B.}$$

După transformări elementare, aceste șase inegalități pot fi scrise:

$$(1_b)\qquad \begin{aligned} -3x_2 + 2x_4 + 2x_5 &\leqslant 0, \\ 7x_2 - 13x_4 + 7x_5 &\leqslant 0, \\ 3x_2 + 3x_4 - 17x_5 &\leqslant 0, \\ -17x_1 + 3x_3 + 3x_6 &\leqslant 0, \\ 9x_1 - 11x_3 + 9x_6 &\geqslant 0, \\ x_1 + x_3 - 3x_6 &\geqslant 0, \end{aligned}$$

Se impun de asemenea și condițiile:

$$(2)\qquad x_i \geqslant 0, \quad i = 1, 2, \ldots, 6.$$

Scanned with OKEN Scanner

Venitul net al intreprinderii este dat de diferența dintre suma încasată în urma vînzării benzinelor A și B și suma cheltuită pentru producerea componentelor B_1, B_2, B_3, B_4 adică

$$z = [270(x_2 + x_4 + x_5) + 320(x_1 + x_3 + x_6)] -$$
$$- [210x_1 + 260(x_2 + x_3) + 240x_4 + 180(x_5 + x_6)].$$

În concluzie, rezolvarea problemei, constă în determinarea valorilor variabilelor x_1, x_2, x_3, x_4, x_5, x_6 și implicit, a cantităților de benzine A, B ce se vor produce, astfel încît valoarea lui z să fie maximă, în condițiile respectării restricțiilor (1_a), (1_b), (2).

Exemplul III (*Planificarea producției*)

O secție a unei uzine fabrică două produse A și B.

Pentru fiecare produs este necesară executarea a trei operații de prelucrare la trei mașini unelte diferite M_1, M_2, M_3.

Se cunosc timpii necesari pentru cele trei operații:

	mașina M_1 *(timp în min)*	*mașina* M_2 *(timp în min)*	*mașina* M_3 *(timp în min)*
produsul A	10	7	5
produsul B	8	11	13

Procesul de producție este condiționat de următoarele norme:

— mașina M_1 poate funcționa lunar (un schimb) un timp de cel mult 160 ore;

— mașina M_2 poate funcționa lunar (un schimb) un timp de cel mult 150 ore;

— mașina M_3 trebuie folosită la maximum, (un schimb) adică 208 ore;

— în producția lunară a secției, produsul B trebuie să aibă o pondere de cel puțin 30%.

Dacă beneficiul realizat dintr-o unitate de produs A, respectiv, B este de 100 lei, respectiv 90 lei, se cere să se planifice producția lunară a secției, astfel ca respectînd normele impuse, să se realizeze beneficii maxime.

Necunoscutele, în cazul acestei probleme, vor fi:

x_1 = numărul de unități din produsul A planificate;

x_2 = numărul de unități din produsul B planificate.

Scanned with OKEN Scanner

Ţinînd cont de timpii necesari pentru prima fază executată la maşina M_1, rezultă că aceasta va trebui să funcţioneze un timp egal cu $(10x_1 + 8x_2)$ minute pentru realizarea producţiei planificate. Analog, timpii de funcţionare ai maşinilor M_2 şi M_3 vor fi $(7x_1 + 11x_2)$ minute, respectiv, $(5x_1 + 13x_2)$ minute.

Impunînd restricţiile privitoare la timpii de funcţionare disponibili, obţinem inecuaţiile

$$(1_a)\qquad \begin{aligned} 10x_1 + 8x_2 &\leqslant 9\,600, \\ 7x_1 + 11x_2 &\leqslant 9\,000, \\ 5x_1 + 13x_2 &= 12\,480. \end{aligned}$$

Producţia lunară totală a atelierului fiind $x_1 + x_2$, problema cere

$$(1_b)\qquad x_2 \geqslant \frac{30}{100}(x_1 + x_2).$$

O cerinţă naturală este şi în acest caz

$$(2)\qquad x_i \geqslant 0, \quad i = 1,2.$$

Beneficiul secţiei va fi dat de expresia

$$z = 100x_1 + 90x_2.$$

Ca atare rezolvarea problemei presupune maximizarea lui z în condiţiile (1_a), (1_b), (2).

Exemplul IV. (*Planificarea producţiei*)

Într-o uzină de produse chimice m articole se pot obţine din n materii prime diferite $M_1, M_2, \ldots, M_n$.

Dintr-o tonă de materie primă M_i se pot produce a_{ij} unităţi de articol $A_j (j = 1, 2, \ldots, m)$.

Conform planului, uzina trebuie să producă lunar, cîte b_j unităţi din fiecare articol $A_j (j = 1, 2, \ldots, m)$.

Să se întocmească planul de consum lunar al materiilor prime astfel ca pentru atingerea producţiei planificate, cheltuielile să fie minime. Se are în vedere că preţul unei tone din materia primă M_i este $c_i (i = 1, 2, \ldots, n)$.

Notăm cu x_i cantitatea de materie primă M_i ce se va folosi în procesul de producţie $(i = 1, 2, \ldots, n)$.

Scanned with OKEN Scanner

Pentru realizarea planului, necunoscutele x_i vor trebui să satisfacă următoarele restricţii:

$$(1) \qquad \sum_{i=1}^{n} a_{ij}x_i \geqslant b_j, \qquad j = 1, 2, \ldots, m;$$

$$(2) \qquad x_i \geqslant 0, \qquad i = 1, 2, \ldots, n.$$

Cheltuielile de producţie, în absenţa altor factori, sînt date de costul materiilor prime folosite. Minimizarea lor implică realizarea cerinţei

$$(3) \qquad z = \sum_{i=1}^{n} c_i x_i = \text{minim.}$$

Exemplul V (*Problema transporturilor*)

m fabrici $F_1, F_2, \ldots, F_m$ aprovizionează cu acelaş produs, n localităţi $D_1, D_2, \ldots, D_n$.

Capacităţile lunare de producţie, pentru produsul respectiv sînt egale cu $a_i (i = 1, 2, \ldots, m)$ (pentru fabrica F_i).

Necesarul, pentru localitatea D_j, pentru acelaş produs este b_j $(j = 1, 2, \ldots, n)$.

Cheltuielile pentru transportul unei unităţi de produs de la fabrica F_i la localitatea D_j fiind c_{ij} $(i = 1, 2, \ldots, m; \; j = 1, 2, \ldots, n)$ se pune problema satisfacerii cererilor în cele n localităţi, astfel încît cheltuielile totale de transport să fie minime

Presupunem $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$.

Să notăm cu x_{ij} cantitatea de produs ce se va repartiza de la sursa i (adică fabrica F_i) la destinaţia j (localitatea D_j)

Cantitatea de produs care ajunge la localitatea D_j este suma cantităţilor trimise aici de fiecare din cele m surse, adică

$$\sum_{i=1}^{m} x_{ij}.$$

Pentru ca toate cererile să fie acoperite, impunem restricţiile

$$(1_a) \qquad \sum_{i=1}^{m} x_{ij} = b_j, \qquad j = 1, 2, \ldots, n.$$

Scanned with OKEN Scanner

Cantitatea de produs care pleacă de la sursa i se obţine sumînd cantităţile expediate de la aceasta către cele n destinaţii, fiind deci egală cu

$$\sum_{j=1}^{n} x_{ij} .$$

Evident, pentru fiecare sursă, această cantitate trebuie să coincidă cu cantitatea produsă aici, deci

$$(1_b) \qquad \sum_{j=1}^{n} x_{ij} = a_i, \qquad i = 1, 2, \ldots, m.$$

Şi în acest caz se impun condiţiile

$$(2) \qquad x_{ij} \geqslant 0, \quad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n.$$

Pentru calcularea preţului total al transporturilor, trebuie făcută suma produselor dintre cantităţile expediate de la sursa i la destinaţia j şi preţurile unitare de transport între cele două puncte.

Condiţia de minimizare a cheltuielilor de transport se scrie deci

$$(3) \qquad z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} = \text{minim.}$$

Exemplul VI.

Trei ferme F_1, F_2, F_3 au planificat să însămînţeze suprafeţele arabile de care dispun cu patru tipuri de culturi C_1, C_2, C_3, C_4.

Suprafeţele disponibile ale fiecărei ferme sînt de 150 ha, 80 ha, respectiv, 200 ha, iar cele prevăzute a fi cultivate, pe ansamblul celor trei ferme sînt: 100 ha pentru prima cultură, 100 ha pentru cea de a doua, 140 ha respectiv, 90 ha pentru celelalte.

Timpii de muncă (în om/zi), calculaţi pentru fiecare fermă şi cultură în parte sînt cunoscuţi şi daţi în următorul tablou:

cultura / *ferma*	C_1	C_2	C_3	C_4
F_1	1	3	5	2
F_2	1,5	2	4	2
F_3	2	4	3	1

Scanned with OKEN Scanner

Să se repartizeze terenul arabil al fiecărei ferme între cele patru culturi, de aşa manieră încît, realizînd planul fixat pe ansamblu, să se presteze un timp de muncă minim.

Pentru modelarea problemei notăm cu x_{ij} aria suprafeţei (în ha) aparţinînd fermei F_i, $(i = 1, 2, 3)$ destinată culturii C_j, $(j = 1, 2, 3, 4)$.

Pentru ca sarcina de plan să fie realizată trebuie ca

$$(1_a)\qquad \begin{aligned} x_{11} + x_{21} + x_{31} &= 100,\\ x_{12} + x_{22} + x_{32} &= 100,\\ x_{13} + x_{23} + x_{33} &= 140,\\ x_{14} + x_{24} + x_{34} &= 90. \end{aligned}$$

Suprafeţele fiecărei ferme sînt complet utilizate dacă

$$(1_b)\qquad \begin{aligned} x_{11} + x_{12} + x_{13} + x_{14} &= 150,\\ x_{21} + x_{22} + x_{23} + x_{24} &= 80,\\ x_{31} + x_{32} + x_{33} + x_{34} &= 200. \end{aligned}$$

Adăugăm şi condiţiile

$$(2)\qquad x_{ij} \geqslant 0, \qquad i = 1, 2, 3;\ j = 1, 2, 3, 4.$$

Timpul de muncă prestat pe ansamblul fermelor este

$$z = x_{11} + 3x_{12} + 5x_{13} + 2x_{14} + 1{,}5x_{21} + 2x_{22} + 4x_{23} + 2x_{24} + \\ + 2x_{31} + 4x_{32} + 3x_{33} + x_{34}.$$

Problema cere

$$(3)\qquad z = \text{minim}.$$

Ultimile două probleme, sub formă matematică, sînt identice şi le vom încadra în acelaşi timp, denumit curent, „problema transporturilor”

În toate cele şase exemple, în pofida unor diferenţe nete sub aspectul concret, reţinem o serie de asemănări în modelul matematic, constînd, în esenţă în:

— existenţa unui grup de restricţii (1), reprezentat printr-un sistem de ecuaţii sau inecuaţii liniare.

Scanned with OKEN Scanner

— existența unui grup de restricții (2) care specifică nenegativitatea necunoscutelor.

— condiția de optim se referă la o expresie liniară în necunoscutele problemei.

Aceste trăsături comune constituie motivul pentru care, asemenea probleme pot fi, matematic integrate unui unic model și rezolvate ca atare.

§ 2. REZOLVAREA GRAFICĂ A UNOR PROBLEME SIMPLE

Problemele de programare liniară în care intervin două sau trei variabile se pot rezolva pe cale grafică.

Interpretarea geometrică a problemelor programării liniare este destul de importantă și vom face uz de ea în cuprinsul lucrării. Pe exemplele care urmează vom putea trage unele concluzii pe care, ulterior, le vom demonstra riguros.

Exemplul I.

Să începem prin a găsi interpretarea geometrică a următoarei probleme:

$$
\begin{aligned}
& & x_2 &\leqslant 10, \\
(1) & & 3x_1 + 2x_2 &\leqslant 30, \\
& & 10x_1 + x_2 &\leqslant 70, \\
(2) & & x_1 \geqslant 0,\ & x_2 \geqslant 0.
\end{aligned}
$$

(3) să se realizeze maximul valorilor $z = 15x_1 + 5x_2$.

Putem atașa fiecărei perechi de valori (x_1, x_2), ale variabilelor problemei, un punct în planul raportat unui sistem de axe rectangulare.

Într-o primă etapă ne propunem să determinăm mulțimea tuturor punctelor din acest plan, ale căror coordonate x_1 și x_2 verifică condițiile (1) și (2).

Geometric, fiecare inecuație din grupurile (1) și (2) determină în plan un semiplan, și anume, unul din cele două semiplane în care este divizat planul de către dreapta a cărei ecuație se obține transformînd inegalitatea respectivă în ecuație (fig. 3)

Scanned with OKEN Scanner

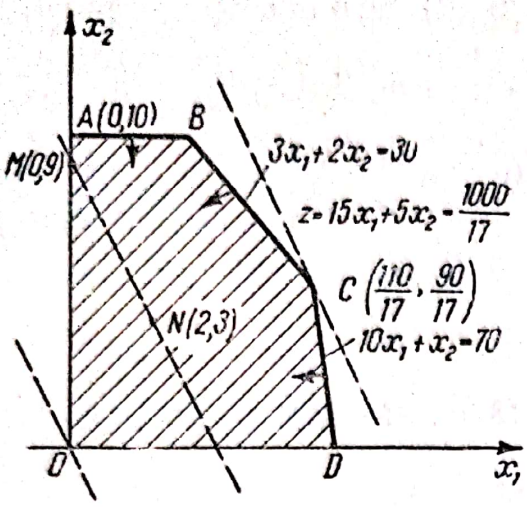


Fig. 3.

În exemplul considerat, inegalitatea $x_2 \leqslant 10$ este verificată de toate punctele situate în semiplanul din stînga dreptei de ecuaţie $x_2 = 10$. Următoarea inecuaţie, $3x_1 + 2x_2 \leqslant 30$, reprezintă semiplanul determinat de dreapta de ecuaţie $3x_1 + 2x_2 = 30$, şi care este marcat cu ajutorul unei săgeţi în figura 3. Analog, pentru inecuaţia $10x_1 + 2x_2 \leqslant 70$, am pus în evidenţă pe figură unul din semiplanele determinate de dreapta de ecuaţie

$$10x_1 + x_2 = 70.$$

Condiţiile (2) ne limitează la cadranul I.

În definitiv, mulţimea tuturor punctelor ale căror coordonate verifică condiţiile (1) şi (2), şi pe care le vom numi *soluţii posibile ale problemei*, se vor găsi în poligonul $OABCD$.

Fie acum $z = 15x_1 + 5x_2$. Geometric, această egalitate reprezintă pentru fiecare z fixat, o dreaptă. Astfel se obţine o familie de drepte paralele de pantă -3. În figura 3 am trasat trei drepte ale familiei, corespunzînd unor valori diferite ale lui z.

Un rezultat binecunoscut permite evaluarea distanţei de la un punct la o dreaptă, în particular de la punctul $O(0, 0)$ la dreapta $15x_1 + 5x_2 = z$.

$$d = \frac{15 \times 0 + 5 \times 0 - z}{-\sqrt{225 + 25}} = \frac{z}{\sqrt{250}}.$$

Această relaţie simplă sugerează proporţionalitatea dintre z şi distanţa la origine a dreptei.

Cerinţa (3) a problemei este satisfăcută dacă sînt găsite valorile x_1 şi x_2 pentru care expresia $15x_1 + 5x_2$ este maximă.

Ca atare, soluţia optimă a problemei trebuie să corespundă unui punct (x_1, x_2) care să se găsească în poligonul $OABCD$ (pentru a fi îndeplinite condiţiile (1) şi (2)) şi care, în acelaşi timp, se găseşte pe o dreaptă de forma $15x_1 + 5x_2 = z$ în care z este maxim posibil. Adică, să fie punct comun al poligonului şi al dreptei celei mai îndepărtate de origine.

Scanned with OKEN Scanner

Pe figura 3 se vede că $x_1 = 0$, $x_2 = 9$ este o soluție posibilă deoarece punctul $M(0, 9)$ este în interiorul poligonului $OABCD$. Valoarea expresiei $(15x_1 + 5x_2)$ pentru acest punct este 45, aceeași ca și pentru punctul $N(2, 3)$ care corespunde unei alte soluții posibile $x_1 = 2$, $x_2 = 3$. Aceasta este, de altfel evident, deoarece cele două puncte se găsesc pe aceeași dreaptă $15x_1 + 5x_2 = 45$.

Dreapta familiei cea mai depărtată de origine și care are puncte comune cu poligonul soluțiilor posibile este $15x_1 + 5x_2 = \frac{1000}{17}$.

Punctul comun este tocmai vîrful C al poligonului și deci, coordonatele sale $x_1 = \frac{110}{17}$, $x_2 = \frac{90}{17}$ constituie soluția optimă a problemei.

Valoarea $z = \frac{1000}{17}$ este superioară oricărei alte valori pe care o poate lua z în punctele din poligonul $OABCD$.

Exemplul II

$$
\begin{aligned}
&4x_1 + 3x_2 \geqslant 24,\\
&5x_1 + 9x_2 \geqslant 45,\\
&-3x_1 + 4x_2 \leqslant 44,\\
&x_1 \leqslant 8;
\end{aligned}
\qquad (1)
$$

$$
x_1 \geqslant 0,\ x_2 \geqslant 0; \qquad (2)
$$

(3) să se realizeze minimul valorilor $z = x_1 + x_2$.

Domeniul în care $4x_1 + 3x_2 \geqslant 24$, este semiplanul determinat de dreapta $4x_1 + 3x_2 = 24$ care nu conține originea; cel pentru care $5x_1 + 9x_2 \geqslant 45$ este semiplanul determinat de dreapta de ecuație $5x_1 + 9x_2 = 45$ care nu conține originea. Tot astfel, regiunile din plan pentru care se verifică $-3x_1 + 4x_2 \leqslant 44$ și $x_1 \leqslant 8$ sînt semiplanele determinate de dreptele $-3x_1 + 4x_2 = 44$ și respectiv, $x_1 = 8$ care conțin originea.

Dacă ținem cont și de (2), rezultă că mulțimea soluțiilor posibile este reprezentată de poligonul $ABCDE$ (fig. 4).

Pe baza observației făcute în exemplul I, valoarea lui z fiind proporțională cu distanța de la origine la dreapta $x_1 + x_2 = z$,

Scanned with OKEN Scanner

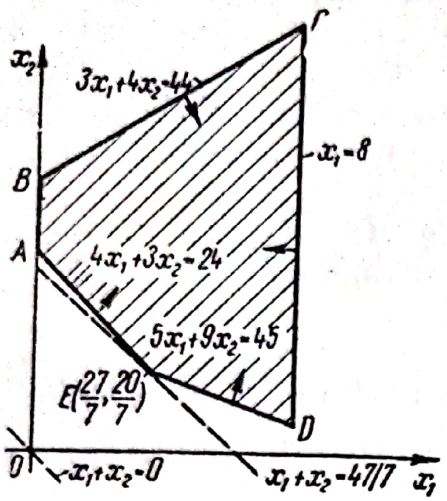


Fig. 4

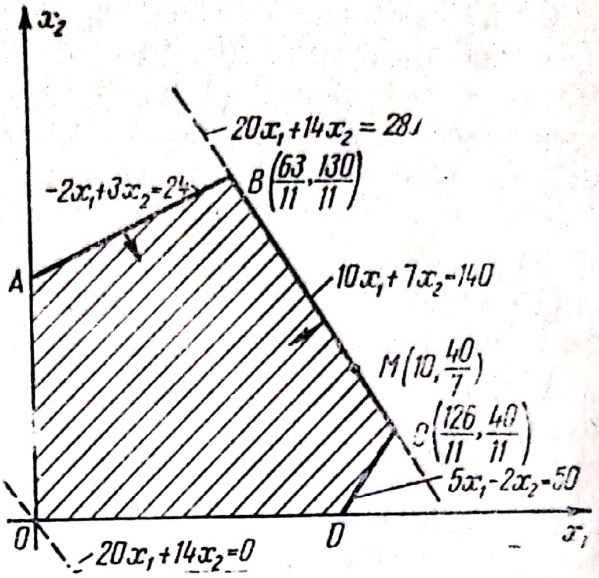


Fig. 5

decidem că soluţia optimă va corespunde punctului din poligonul $ABCDE$ aflat pe dreapta din familia $x_1 + x_2 = z$ cea mai apropiată de origine. Adică $E(27/7,\ 20/7)$. Deci $x_1 = 27/7$, $x_2 = 20/7$ este soluţia optimă.

Exemplul III.

$$
\begin{aligned}
-\ 2x_1 + 3x_2 &\leqslant 24,\\
5x_1 - 2x_2 &\leqslant 50,\\
10x_1 + 7x_2 &\leqslant 140,\\
x_1 \geqslant 0,\quad x_2 &\geqslant 0\,;
\end{aligned}
\qquad (1)\qquad (2)
$$

(3) să se realizeze maximul valorilor $z = 20x_1 + 14x_2$.

Domeniul corespunzător soluţiilor posibile se obţine pe cale grafică ca în exemplele anterioare, rezultînd poligonul $OABCD$ (fig. 5)

Să examinăm acum, dreptele familiei $20x_1 + 14x_2 = z$.

Panta acestor drepte este $m = -\frac{20}{14}$ şi coincide cu cea a uneia din dreptele care delimitează poligonul soluţiilor posibile, anume, cu BC.

Deoarece (3) cere realizarea unui maxim, sîntem conduşi la alegerea acelei drepte a familiei care avînd puncte comune cu poligonul $OABCD$, se află la distanţă maximă de origine.

Scanned with OKEN Scanner

Or, această dreaptă, $20x_1 + 14x_2 = 280$ se suprapune peste segmentul BC. Vor exista deci, o infinitate de soluţii optime ale problemei : toate perechile x_1, x_2 care dau coordonatele punctelor din segmentul BC.

Exemplul IV

În exemplele precedente, mulţimile soluţiile posibile erau infinite, reprezentîndu-se geometric prin cîte un întreg poligon convex. Pentru soluţia optimă se punea deci problema alegerii acelui, sau acelor puncte din mulţimea respectivă care să răspundă şi la cerinţa de optim a problemei.

Fie acum problema :

$$
\begin{aligned}
&3x_1 + 7x_2 \geqslant 21,\\
&2x_1 - x_2 \leqslant 14,\\
(1)\qquad &6x_1 + 2x_2 = 18,\\
&x_1 + x_2 = 6;\\
(2)\qquad &x_1 \geqslant 0,\quad x_2 \geqslant 0,
\end{aligned}
$$

(3) să se minimizeze expresia $z = 6x_1 - 2x_2$.

Inegalităţile $3x_1 + 7x_2 \geqslant 21$ şi $2x_1 - x_2 \leqslant 14$ împreună cu (2) determină în plan zona care apare haşurată în figura 6. Ţinînd seama şi de cele două egalităţi din (1), mulţimea soluţiilor posibile se reduce la unicul punct $C(3/2,\ 9/2)$.

Bineînţeles, în acest caz nu se mai pune problema alegerii soluţiei posibile care să minimizeze pe z. Unica soluţie posibilă, şi deci, şi cea optimă este $x_1 = 3/2$, $x_2 = 9/2$ pentru care $z = 0$.

Exemplul V.

$$
\begin{aligned}
&4x_1 + 5x_2 \geqslant 20,\\
(1)\qquad &7x_1 + 2x_2 \geqslant 14;\\
(2)\qquad &x_1 \geqslant 0,\ x_2 \geqslant 0;\\
(3)\qquad &4x_1 + 9x_2 = z = \text{maxim.}
\end{aligned}
$$

Domeniul soluţiilor posibile apare haşurat în figura 7.
Caracteristica acestui domeniu este de a fi nemărginit.

Scanned with OKEN Scanner

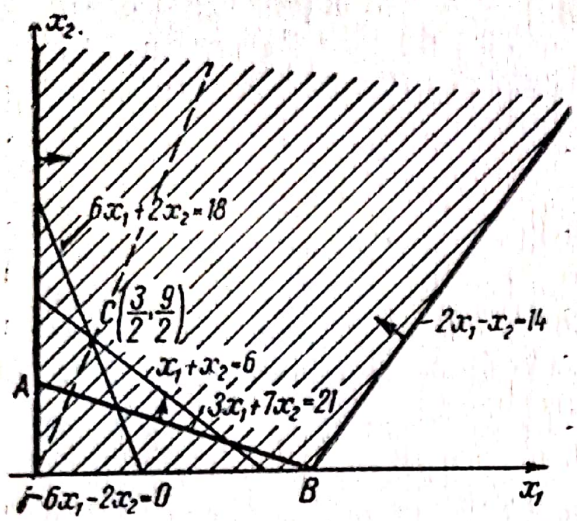


Fig 6

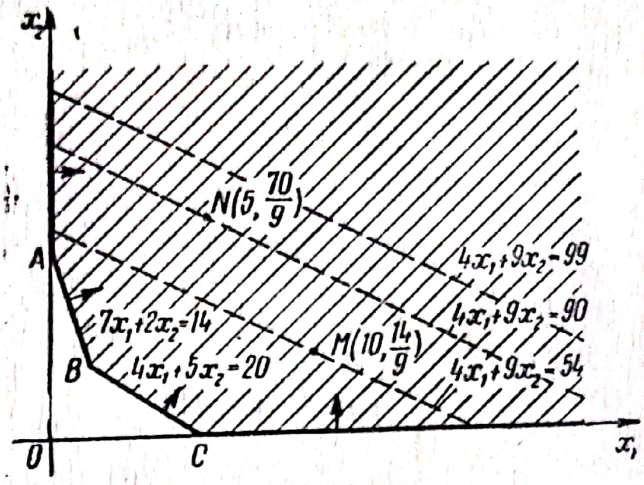


Fig. 7

Pentru realizarea condiției de maxim (3), ar trebui căutată dreapta din familia $4x_1 + 9x_2 = z$, care, intersectînd domeniul x_2ABCx_1, să realizeze distanța maximă de origine. Or, se poate constata că de la o anumită valoare a lui z, orice dreaptă a familiei intersectează domeniul soluțiilor posibile.

Deci fiecare nouă poziție mai depărtată de origine aduce o majorare a expresiei $4x_1 + 9x_2$.

Cu alte cuvinte, soluția poate fi necontenit îmbunătățită, fără a exista un prag superior care să limiteze procesul.

Problema de programare liniară nu are, în acest caz, soluție optimă finită, deoarece cel puțin una din componentele sale, x_1 sau x_2, poate fi necontenit mărită.

Exemplul VI.

$$
\begin{aligned}
&9x_1 + 30x_2 \leqslant 210,\\
&4x_1 + 11x_2 \geqslant 44,\\
&6x_1 + 4x_2 \geqslant 24,\\
&3x_1 + 10x_2 \leqslant 30;
\end{aligned}
\tag{1}
$$

$$x_1 \geqslant 0, \quad x_2 \geqslant 0; \tag{2}$$

(3) să se minimizeze $z = 3x_1 + 4x_2$.

Scanned with OKEN Scanner

După cum se remarcă pe figura 8, domeniul determinat de condiţiile $6x_1 + 4x_2 \geqslant 24$ şi $3x_1 + 10x_2 \leqslant 30$, adică triunghiul ABC nu are niciun punct comun cu domeniul EDF determinat de inegalităţile $9x_1 + 30x_2 \leqslant 210$ şi $4x_1 + 11x_2 \geqslant 44$.

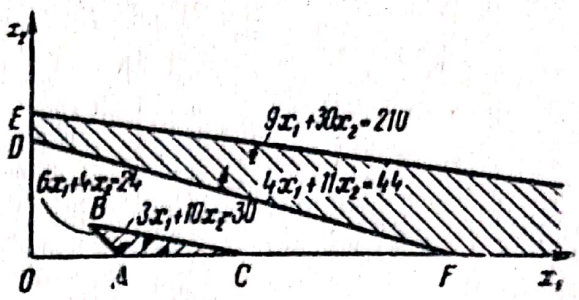


Fig. 8

Această constatare conduce la concluzia că problema nu are soluţii posibile şi implicit, nici soluţii optime.

Exemplul VII.

De data aceasta vom considera o problemă cu trei variabile. Reprezentarea grafică se va face, firește, în spaţiul tridimensional. Să introducem un sistem de axe rectangulare $Ox_1x_2x_3$ şi fie problema:

$$\begin{aligned} 12x_1 + 28x_2 + 7x_3 &\leqslant 84, \\ 8x_1 + 5x_2 + 10x_3 &\leqslant 40, \\ 21x_1 + 7x_2 + 6x_3 &\leqslant 42; \end{aligned} \tag{1}$$

$$x_1 \geqslant 0, \quad x_2 \geqslant 0, \quad x_3 \geqslant 0; \tag{2}$$

(3) să se maximizeze $z = x_1 + x_2 + x_3$.

Inegalitatea $12x_1 + 28x_2 + 7x_3 \leqslant 84$ este satisfăcută de toate punctele dintr-unul din cele două semispaţii determinate de planul $12x_1 + 28x_2 + 7x_3 = 84$. Cum pentru punctul $O(0, 0, 0)$ inegalitatea este satisfăcută, alegerea va cădea asupra semispaţiului care conţine originea. Raţionînd la fel şi pentru celelalte două inegalităţi din (1) şi ţinînd seama şi de (2) rezultă că domeniul soluţiilor posibile este poliedrul $OABCDEFG$ (fig. 9). Pentru atingerea maximului, ţinînd seama de proporţionalitatea dintre valoarea lui z şi depărtarea planului $x_1 + x_2 + x_3 = z$ de origine, se obţine punctul G care aparţine atît poliedrului soluţiilor posibile cît şi planului situat la distanţa maximă de origine.

Scanned with OKEN Scanner

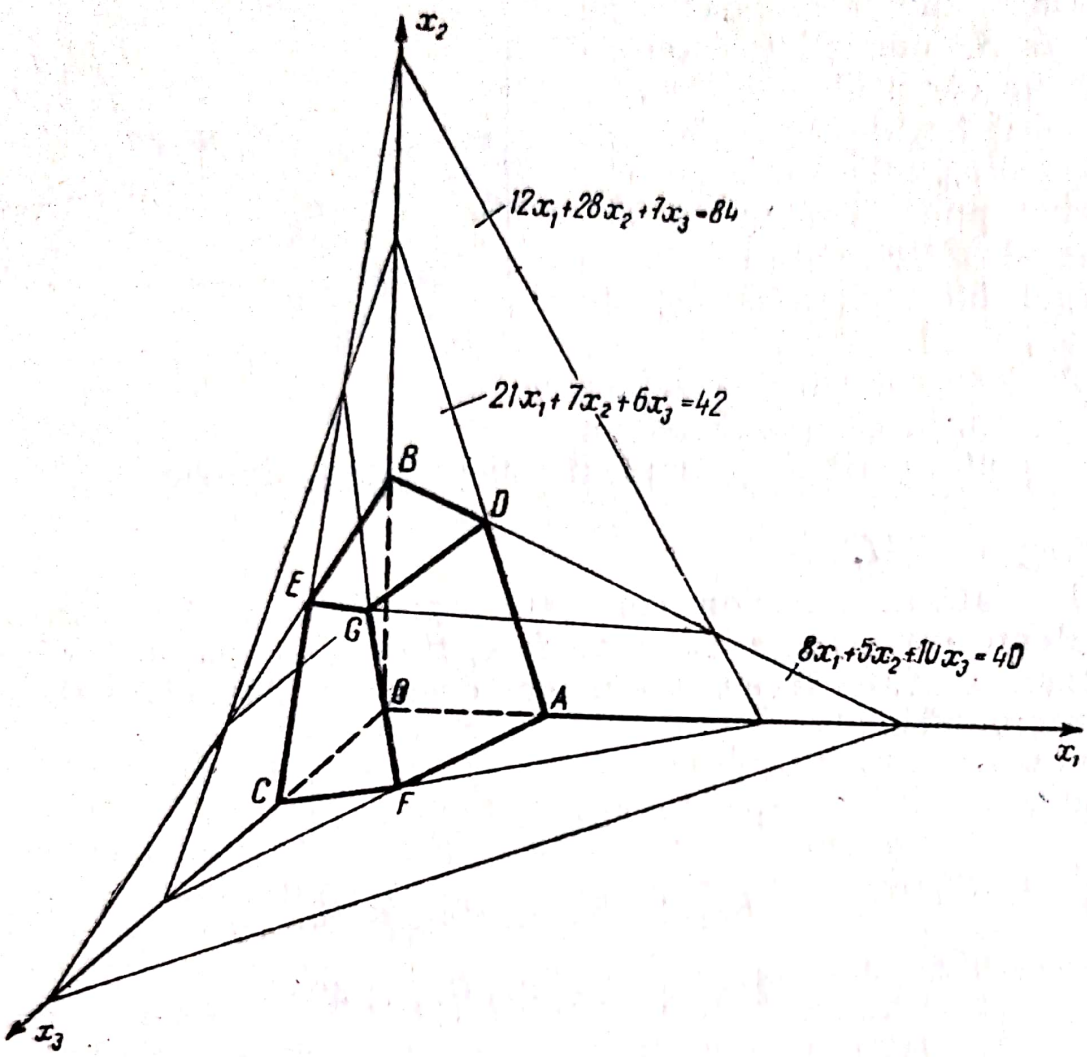


Fig. 9

§ 3 FORMA MATEMATICĂ A PROBLEMEI

3.1. Diverse forme ale problemelor de programare liniară

În § 1, am interpretat matematic cîteva probleme economice concrete. Toate modelele matematice atașate acestora se caracterizează printr-o serie de asemănări structurale, fapt care permite înglobarea lor într-un singur model, general, — al programării liniare.

Forma generală a unei probleme de programare liniară (sau pe scurt, *program liniar*) este:

Să se determine valorile necunoscutelor $x_1, x_2, \ldots, x_n$, care satisfac condiţiile:

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1,$$

$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2,$$

$$\ldots\ldots\ldots\ldots\ldots\ldots$$

$$a_{l1}x_1 + a_{l2}x_2 + \ldots + a_{ln}x_n = b_l,$$

$$a_{l+1,1}x_1 + a_{l+1,2}x_2 + \ldots + a_{l+1,n}x_n \leqslant b_{l+1},$$

(1) $$\ldots\ldots\ldots\ldots\ldots\ldots$$

$$a_{l+s,1}x_1 + a_{l+s,2}x_2 + \ldots + a_{l+s,n}x_n \leqslant b_{l+s},$$

$$a_{l+s+1,1}x_1 + a_{l+s+1,2}x_2 + \ldots + a_{l+s+1,n}x_n \geqslant b_{l+s+1},$$

$$\ldots\ldots\ldots\ldots\ldots\ldots$$

$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \geqslant b_m,$$

$(l \geqslant 1,\ s \geqslant 1,\ l + s \leqslant m)$;

(2) $x_{i_1} \geqslant 0, \quad x_{i_2} \geqslant 0, \ \ldots, \ x_{i_p} \geqslant 0,$

$x_{i_{p+1}} \leqslant 0, \ x_{i_{p+2}} \leqslant 0, \ \ldots, \ x_{i_{p+r}} \leqslant 0, \ p < r \leqslant n,$ restul variabilelor avînd semn nespecificat;

(3) să se maximizeze (minimizeze) forma liniară

$$f(x_1, x_2, \ldots, x_n) = c_1x_1 + c_2x_2 + \ldots + c_nx_n.$$

Reordonînd termenii sistemului de ecuaţii şi inecuaţii liniare (1), pentru a grupa cele p variabile nenegative, respectiv, cele

Scanned with OKEN Scanner

$(r-p)$ variabile nepozitive, se obține o partiționare a matricei coeficienților de forma

$$\left\|\begin{array}{llll|l|l}
a_{1i_1} & a_{1i_2} & \dots & a_{1i_p} & a_{1i_{p+1}} \dots a_{1i_r} & a_{1i_{r+1}} \dots a_{1_{in}} \\
a_{2i_1} & a_{2i_2} & \dots & a_{2i_p} & a_{2i_{p+1}} \dots a_{2i_r} & a_{2i_{r+1}} \dots a_{2i_n} \\
\dots & \dots & \dots & \dots & \dots & \dots \\
a_{li_1} & a_{li_2} & \dots & a_{2i_p} & a_{li_{p+1}} \dots a_{li_r} & a_{li_{r+1}} \dots a_{li_n} \\
\dots & \dots & \dots & \dots & \dots & \dots \\
a_{l+1,i_1} & a_{l+1,i_2} & \dots & a_{l+1,i_p} & a_{l+1,i_{p+1}} \dots a_{l+1,i_r} & a_{l+1,i_{r+1}} \dots a_{l+1,i_n} \\
\dots & \dots & \dots & \dots & \dots & \dots \\
a_{l+s,i_1} & a_{l+s,i_2} & \dots & a_{l+s,i_p} & a_{l+s,i_{p+1}} \dots a_{l+s,i_r} & a_{l+s,i_{r+1}} \dots a_{l+s,i_n} \\
\dots & \dots & \dots & \dots & \dots & \dots \\
a_{l+s+1,i_1} & a_{l+s+1,i_2} & \dots & a_{l+s+1,i_p} & a_{l+s+1,i_{p+1}} \dots a_{l+s+1,i_r} & a_{l+s+1,i_{r+1}} \dots a_{l+s+1,i_n} \\
\dots & \dots & \dots & \dots & \dots & \dots \\
a_{mi_1} & a_{mi_2} & \dots & a_{mi_p} & a_{mi_{p+1}} \dots a_{mi_r} & a_{mi_{r+1}} \dots a_{mi_n}
\end{array}\right\| =$$

$$= \left\| \begin{array}{lll} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{array} \right\| .$$

Cu notațiile

$$X_1 = \left\| \begin{array}{l} x_{i_1} \\ x_{i_2} \\ \vdots \\ x_{i_p} \end{array} \right\|, \quad X_2 = \left\| \begin{array}{l} x_{i_{p+1}} \\ x_{i_{p+2}} \\ \vdots \\ x_{i_r} \end{array} \right\|, \quad X_3 = \left\| \begin{array}{l} x_{i_{r+1}} \\ x_{i_{r+2}} \\ \vdots \\ x_{i_n} \end{array} \right\| ;$$

$$b^1 = \left\| \begin{array}{l} b_1 \\ b_2 \\ \vdots \\ b_l \end{array} \right\|, \quad b^2 = \left\| \begin{array}{l} b_{l+1} \\ b_{l+2} \\ \vdots \\ b_{l+s} \end{array} \right\|, \quad b^3 = \left\| \begin{array}{l} b_{l+s+1} \\ b_{l+s+2} \\ \vdots \\ b_m \end{array} \right\| ;$$

$$C_1 = (c_{i_1}, c_{i_2}, \dots, c_{i_p}), \quad C_2 = (c_{i_{p+1}}, c_{i_{p+2}}, \dots, c_{i_r}),$$

$$C_3 = (c_{i_{r+1}}, c_{i_{r+2}}, \dots, c_{i_n})$$

Scanned with OKEN Scanner

problema precedentă se scrie matriceal

$$A_{11}X_1 + A_{12}X_2 + A_{13}X_3 = b^1,$$

(1′) $$A_{21}X_1 + A_{22}X_2 + A_{23}X_3 \leqslant b^2,$$

$$A_{31}X_1 + A_{32}X_2 + A_{33}X_3 \geqslant b^3;$$

(2′) $$X_1 \geqslant 0, \qquad X_2 \leqslant 0;$$

(3′) să se maximizeze (minimizeze)

$$f(X_1, X_2, X_3) = C_1X_1 + C_2X_2 + C_3X_3.$$

În terminologia curentă, condițiile (1) și (2) se numesc restricții, iar funcționala f, *funcție de eficiență* sau *obiectiv*.

Prin transformări elementare, o problemă de programare liniară poate fi adusă la o formă care, din punct de vedere matematic, se pretează mai bine diverselor metode de rezolvare.

Vom numi *formă standard* a programului liniar prezentarea sa sub următorul aspect:

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1,$$

$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2,$$

(1_s) $$\ldots\ldots\ldots\ldots\ldots\ldots$$

$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m;$$

(2_s) $$x_1 \geqslant 0,\ x_2 \geqslant 0,\ \ldots,\ x_n \geqslant 0;$$

(3_s) să se maximizeze (minimizeze) funcția de eficiență

$$f(x_1, x_2, \ldots, x_n) = c_1x_1 + c_2x_2 + \ldots + c_nx_n$$

sau, scris matriceal,

($1'_s$) $$AX = b;$$

($2'_s$) $$X \geqslant 0;$$

($3'_s$) să se maximizeze (minimizeze) $f(X) = CX$

unde A este matricea coeficienților sistemului (1_s), b este vectorul coloană al termenilor liberi, X vectorul-coloană format cu cele n necunoscute, iar C este vectorul ale cărui componente dau coeficienții necunoscutelor în forma liniară de la (3_s).

Scanned with OKEN Scanner

Prin partiționarea matricei A, după coloanele sale $P_1, P_2, \ldots, P_n$, se obține o formă echivalentă pentru condiția (1_s), numită forma vectorială,

$$x_1P_1 + x_2P_2 + \ldots + x_nP_n = P_0, \quad \text{(unde } P_0 = b\text{)}.$$

Forma canonică a problemei de programare liniară cere determinarea necunoscutelor $x_1, x_2, \ldots, x_n$ satisfăcînd condițiile

$$\begin{aligned} &a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leqslant b_1, \\ &a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leqslant b_2, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \leqslant b_m; \end{aligned} \tag{1_c}$$

$$x_1 \geqslant 0, \; x_2 \geqslant 0, \; \ldots, \; x_n \geqslant 0; \tag{2_c}$$

(3_c) să se maximizeze

$$f(x_1, x_2, \ldots, x_n) = c_1x_1 + c_2x_2 + \ldots + c_nx_n.$$

Cu notațiile matriceale introduse, problema capătă aspectul:

$$AX \leqslant b; \tag{$1'_c$}$$

$$X \geqslant 0. \tag{$2'_c$}$$

$(3'_c)$ să se maximizeze

$$f(X) = CX.$$

Forma canonică se poate scrie și cu inegalități de sens contrar ($\geqslant$) în condiția (1_c) respectiv $(1'_c)$, însă în acest caz (3_c), $(3'_c)$ vor cere minimizarea funcționalei.

Studierea problemei de programare pe forma standard sau canonică, nu înseamnă o restrîngere a generalității ei deoarece, după cum vom vedea, orice problemă de acest tip poate fi adusă la una din formele particulare specificate.

3.2. Reducerea unei probleme de programare liniară la forma standard

Prelucrînd în mod convenabil expresiile ce formează cerințele unei probleme de programare liniară, se poate obține pentru aceasta forma standard. Transformările pe care le vom aplica

Scanned with OKEN Scanner

restricţiilor şi funcţiei obiectiv, trebuie alese de aşa manieră încît ele să nu modifice soluţia problemei.

Să considerăm problema, sub forma generală, aşa cum este enunţată la începutul acestui paragraf.

Pentru aducerea sa la forma standard, este necesar ca toate restricţiile din (1) să fie ecuaţii.

Vom înlocui sistemul format din cele s inecuaţii centrale din (1) (de la $l+1$ la $l+s$) printr-un sistem de ecuaţii liniare echivalent, obţinut prin adăogarea la membrul stîng al fiecărei ecuaţii a unei noi variabile auxiliare nenegative y_i $(i = l+1, l+2, \ldots, l+s)$ (vezi, cap. I, § 5).

Analog, sistemul format din ultimele $m - s - l$ inecuaţii din (1) va fi transformat într-un sistem de ecuaţii în care în afară de necunoscutele $x_1, \ldots, x_n$ apar şi noile variabile auxiliare nenegative y_i $(i = l+s+1, l+s+2, \ldots, m)$, ce se scad din membrul întîi al fiecărei inecuaţii.

Deoarece în forma standard o problemă de programare liniară pretinde numai variabile nenegative, vor trebui făcute unele schimbări de variabile convenabile astfel ca în noua formă problema considerată să conţină numai variabile ce verifică această condiţie.

Conform condiţiilor (2), variabilele $x_{i_1}, x_{i_2}, \ldots, x_{i_p}$ sînt nenegative, ca atare vor rămîne în aceiaşi formă.

Întrucît $x_{i_{p+1}}, x_{i_{p+2}}, \ldots, x_{i_n}$ sînt negative, introducem noi variabile nenegative prin substituţiile

$$w_k = -x_k, \quad (k = i_{p+1}, i_{p+2}, \ldots, i_n).$$

În sfîrşit, variabilele x_k $(k = i_{r+1}, i_{r+2}, \ldots, i_n)$, care nu au semn specificat în (2), se pot înlocui fiecare cu diferenţa a două variabile u_k şi v_k presupuse nenegative. Deci, se efectuează substituţiile

$$x_k = u_k - v_k, \quad u_k \geqslant 0, \quad v_k \geqslant 0, \quad (k = i_{r+1}, i_{r+2}, \ldots, i_n).$$

Operînd aceste substituţii în grupul condiţiilor (1) şi în expresia funcţionalei liniare, se obţine o nouă problemă în necunoscutele $x_{i_1}, x_{i_2}, \ldots, x_{i_p}$; $w_{i_{p+1}}, w_{i_{p+2}}, \ldots, w_{i_n}$;

$u_{i_{r+1}}, u_{i_{r+2}}, \ldots, u_{i_n}, v_{i_{r+1}}, v_{i_{r+2}}, \ldots, v_{i_n}$; $y_{l+1}, y_{l+2}, \ldots, y_m$.

Scanned with OKEN Scanner

Observăm deci, că noua problemă, cere determinarea unui număr de $r + 2(n - r) + (m - l) = 2n + m - r - l$ necunoscute nenegative, supuse unor restricţii ce se exprimă sub forma unor ecuaţii liniare, şi care trebuie să optimizeze o funcţională liniară.

Exemplu.

Ne propunem să reducem la forma standard următoarea problemă:

$$\begin{aligned}
3x_1 + x_2 - x_3 + 2x_4 + x_5 &= 9,\\
x_1 - 2x_2 + x_3 - x_5 + 2x_6 + x_7 &\leqslant 6,\\
-x_1 + 3x_2 - x_3 + 3x_4 + 2x_5 + 5x_6 + x_7 &\geqslant 3,\\
2x_1 - x_2 + 2x_3 - x_4 - x_5 + 8x_6 + 5x_7 &\geqslant 6,\\
x_1 \geqslant 0, \quad x_2 &\geqslant 0,\\
x_4 \leqslant 0, \quad x_6 \leqslant 0, \quad x_7 &\leqslant 0;
\end{aligned}$$

x_3 şi x_5 fără restricţii de semn.

Să se maximizeze funcţia

$$f(X) = 3x_1 + 2x_2 - x_3 - x_4 + x_6 - x_7.$$

Rezolvare.

Pentru cele trei variabile negative se vor face substituţiile

$$w_4 = -x_4, \quad w_6 = -x_6, \quad w_7 = -x_7.$$

Înlocuim variabilele x_3 şi x_5 cu variabilele u_3, v_3, u_5, v_5, după formulele

$$x_3 = u_3 - v_3, \quad x_5 = u_5 - v_5.$$

Operăm toate aceste substituţii în sistemul iniţial concomitent cu transformarea inecuaţiilor în ecuaţii:

$$\begin{aligned}
3x_1 + x_2 - (u_3 - v_3) - 2w_4 + (u_5 - v_5) &= 9,\\
x_1 - 2x_2 + (u_3 - v_3) - (u_5 - v_5) - 2w_6 - w_7 + y_1 &= 6,\\
-x_1 + 3x_2 - (u_3 - v_3) - 3w_4 + 2(u_5 - v_5) - 5w_6 - w_7 - y_2 &= 3,\\
2x_1 - x_2 + 2(u_3 - v_3) + w_4 - (u_5 - v_5) - 8w_6 - 5w_7 - y_3 &= 6.
\end{aligned}$$

Scanned with OKEN Scanner

Adusă la forma standard problema se enunţă astfel:

Să se determine valorile necunoscutelor x_1, x_2, u_3, v_3, w_4, u_5, v_5, w_6, w_7 care satisfac condiţiile:

$$\begin{aligned}
3x_1+ x_2- u_3+ v_3-2w_4+ u_5- v_5 &= 9,\\
x_1-2x_2+ u_3- v_3 - u_5+ v_6-2w_6- w_7+y_1 &= 6,\\
-x_1+3x_2- u_3+ v_3-3w_4+2u_5-2v_5-5w_6- w_7 -y_2 &= 3,\\
2x_1- x_2+2u_3-2v_3+ w_4- u_5+ v_5-8w_6-5w_7 -y_3 &= 6;
\end{aligned}$$

$$x_1 \geqslant 0,\ x_2 \geqslant 0,$$

$$w_4 \geqslant 0,\ w_6 \geqslant 0,\ w_7 \geqslant 0,$$

$$u_3 \geqslant 0,\ v_3 \geqslant 0,\ u_5 \geqslant 0,\ v_5 \geqslant 0,$$

$$y_1 \geqslant 0,\ y_2 \geqslant 0,\ y_3 \geqslant 0;$$

$$3x_1 + 2x_2 - u_3 + v_3 + w_4 - w_6 + w_7 = \text{maxim}.$$

Observaţie. valorile necunoscutelor y_1, y_2, y_3 nu prezintă interes pentru problemă, ele neintervenind în cerinţa de optim. Ca atare, nu au fost menţionate în text, ele apărînd doar în sistem cu scopul de a compensa inegalităţile dintre cei doi membri ai fiecărei inecuaţii.

Menţionăm de asemenea că în orice cerinţă de optimizare, maximul şi minimul se pot înlocui reciproc pe baza egalităţilor

$$\max f(X) = -\min(-f(X)),$$
$$\min f(X) = -\max(-f(X)).$$

§ 4. SOLUŢIILE PROBLEMEI PROGRAMĂRII LINIARE

Pe baza rezultatelor precedente, este suficient studiul problemelor de programare liniară în forma standard.

Vom consacra deci acest paragraf, clasificării şi studierii proprietăţilor soluţiilor unei probleme de programare liniară adusă la forma (S):

(1) $$AX = b,$$

(2) $$X \geqslant 0,$$

(3) să se maximizeze $f(X) = CX$

Scanned with OKEN Scanner

unde A este o matrice de tip $(m \times n)$, b un vector coloană $(m \times 1)$, C un vector linie $(1 \times n)$, iar X vectorul coloană $(n \times 1)$ ale cărui componente trebuie determinate.

Presupunem că *rangul matricei A este m*, ipoteză care va rămîne neschimbată atît în acest paragraf cît și în cel următor.

Introducînd unele definiții consacrate desemnînd vectorii X care satisfac parțial sau integral cerințele problemei (S), vom demonstra în continuare proprietăți importante ale soluțiilor acesteia, proprietăți care constituie baza teoretică a metodelor de rezolvare a problemelor de programare liniară.

Totodată vom căuta să dăm și o interpretare geometrică acestor proprietăți, generalizînd și extinzînd astfel, imaginea intuitivă care a fost pusă în evidență de exemplele grafice ale paragrafului 2.

Vom folosi adesea, scrierea condiției (1) sub forma vectorială

$$x_1P_1 + x_2P_2 + \ldots + x_nP_n = P_0, \tag{1'}$$

în care $P_1, P_2, \ldots, P_n$ sînt vectorii coloană $(m \times 1)$, din matricea A, iar $P_0 = b$.

Pentru concizia și claritatea expunerii, vom denumi spațiul vectorial m-dimensional — în care sînt integrați și acești vectori *spațiul restricțiilor*.

În același spirit, spațiul vectorial n-dimensional, în care se găsesc toți vectorii X, va fi denumit *spațiul soluțiilor*.

Definiție. *Se numește soluție posibilă a problemei (S), un vector X din spațiul soluțiilor, ale cărui componente $x_1, x_2, \ldots, x_n$, verifică relațiile* (1) *și* (2) (*sau, echivalent,* (1') *și* (2').).

Întrucît chestiunea existenței și aceea a unicității soluției posibile a problemei (S), se reduc la problemele respective pentru sistemele de ecuații liniare și, respectiv, inecuații liniare (1) și (2), ele nu vor constitui obiectul studiului acestor capitole. În cele ce urmează nu vom ridica problema existenței soluțiilor posibile. Proprietățile referitoare la acestea, așa cum vor fi enunțate în continuare, vor fi valabile în ipoteza existenței unor asemenea soluții.

Mulțimea $\mathfrak{X}$ a soluțiilor posibile va constitui deci, o submulțime a spațiului vectorial n-dimensional al soluțiilor, mulțime ce poate fi vidă, redusă la un punct, infinită dar mărginită, infinită și nemărginită, așa cum se poate remarca ușor din exemplele prezentate în § 2.

Scanned with OKEN Scanner

Teorema 1. *Mulțimea $\mathfrak{X}$ a soluțiilor posibile este convexă.*

Demonstrație. Dacă $\mathfrak{X}$ este vidă, sau este formată dintr-un singur punct, concluzia este banală.

Presupunem deci că $\mathfrak{X}$ nu este vidă și fie X_1 și X_2 două puncte diferite ale sale.

Vom arăta că orice alt punct X al lui $\mathfrak{X}$, de forma

$$X = \lambda X_1 + (1 - \lambda)X_2 \quad \text{cu} \quad 0 \leqslant \lambda \leqslant 1$$

este de asemenea o soluție posibilă a problemei (S).

Într-adevăr,

$$AX = A[\lambda X_1 + (1-\lambda)X_2] = \lambda AX_1 + (1 - \lambda)AX_2,$$

ultima egalitate rezultînd din proprietățile operațiilor cu matrice.

Deoarece X_1 și X_2 sînt soluții posibile, ele verifică condiția (1) și deci

$$AX_1 = b \text{ și } AX_2 = b.$$

Introduse în relația precedentă, aceste egalități dau

$$AX = \lambda b + (1 - \lambda)b = b.$$

Egalitatea $AX = b$ dovedește că X verifică (1).

Pe de altă parte, X_1 și X_2 fiind soluții posibile,

$$X_1 \geqslant 0 \text{ și } X_2 \geqslant 0.$$

Cum $0 \leqslant \lambda \leqslant 1$, înmulțind aceste două inegalități cu numerele nenegative λ, respectiv $(1 - \lambda)$, sensul lor nu va fi modificat și

$$\lambda X_1 \geqslant 0, \qquad (1 - \lambda)X_2 \geqslant 0.$$

Adunate, membru cu membru, aceste relații dau

$$\lambda X_1 + (1 - \lambda)X_2 \geqslant 0$$

și ținînd seama de semnificația primului membru, rezultă.

$$X \geqslant 0$$

adică X verifică și condiția (2), deci este o soluție posibilă, și deci $\mathfrak{X}$ este convexă.

Corolar. *Dacă $\mathfrak{X}$ conține mai mult de un punct, ea conține de fapt o infinitate.*

Scanned with OKEN Scanner

Din definiţie rezultă că componentele $x_1, x_2, \ldots, x_n$ ale soluţiei posibile X sînt nenegative şi satisfac condiţia (1')

$$x_1P_1 + x_2P_2 + \ldots + x_nP_n = P_0.$$

Această relaţie, indică posibilitatea de a scrie vectorul P_0 ca o combinaţie liniară a vectorilor $P_1, P_2, \ldots, P_n$, în care coeficienţii sînt tocmai componentele lui X.

Deoarece rangul matricei A este m, aceasta implică existenţa unui sistem de m vectori $P_i (i = 1, 2, \ldots, n)$, liniar independenţi.

Un asemenea sistem constituie, totodată, o bază în spaţiul vectorial m-dimensional al restricţiilor.

Definiţie. *O soluţie posibilă X, care are cel mult m componente $x_{i_1}, x_{i_2}, \ldots, x_{i_r} (r \leqslant m)$ strict pozitive, va fi numită soluţie de bază, dacă vectorii $P_{i_1}, P_{i_2}, \ldots, P_{i_r}$ sînt liniar independenţi.*

Întrucît condiţia (2) impune componentelor soluţiilor posibile condiţia de nenegativitate, acestea pot fi, evident, strict pozitive sau nule. Definiţia soluţiei de bază, limitează numărul componentelor strict pozitive (deci nenule) la m.

Dacă în soluţia de bază m componente sînt nenule, ea se va numi *nedegenerată*, în caz contrar (dacă va conţine mai puţin de m componente nenule) se va numi *degenerată*.

Din definiţia precedentă, rezultă că nu orice soluţie posibilă cu cel mult m componente nenule este o soluţie de bază.

Se impune de asemenea condiţia de independenţă a vectorilor coloană P_i corespunzători componentelor pozitive.

Să urmărim pe exemple numerice unele aspecte ale acestei probleme.

Exemple.

a)

(1a) $$\begin{aligned} x_1 - x_2 + 2x_3 + 3x_4 &= 5, \\ 2x_1 + 3x_2 + 4x_3 - x_5 &= 10; \end{aligned}$$

(2a) $$x_1 \geqslant 0,\ x_2 \geqslant 0,\ x_3 \geqslant 0,\ x_4 \geqslant 0,\ x_5 \geqslant 0.$$

Matricea $A = \begin{Vmatrix} 1 & -1 & 2 & 3 & 0 \\ 2 & 3 & 4 & 0 & -1 \end{Vmatrix}$ are rangul 2.

Vectorul $X_1 = \begin{Vmatrix} 1 \\ 5 \\ 0 \\ 3 \\ 7 \end{Vmatrix}$, reprezintă o soluţie posibilă a problemei

întrucît verifică condiţiile (1_a) şi (2_a).

Vectorul $X_2 = \begin{Vmatrix} 0 \\ 10/3 \\ 0 \\ 25/9 \\ 0 \end{Vmatrix}$ constituie o soluţie de bază nedegenerată.

Într-adevăr el verifică condiţiile (1a) şi (2a) deci este o soluţie posibilă. De asemenea numărul componentelor nenule coincide cu rangul matricei A, iar vectorii $P_2 = \begin{Vmatrix} -1 \\ 3 \end{Vmatrix}$ şi $P_4 = \begin{Vmatrix} 3 \\ 0 \end{Vmatrix}$ corespunzători acestor componente, sînt liniar independenţi, întrucît determinantul ataşat matricei

$$\begin{Vmatrix} -1 & 3 \\ 3 & 0 \end{Vmatrix}$$

este nenul.

Vectorul $X_3 = \begin{Vmatrix} 0 \\ 0 \\ 5/2 \\ 0 \\ 0 \end{Vmatrix}$ este o soluţie de bază degenerată,

Întrucît singurei componente nenule $x_3 = 5/2$, îi corespunde vectorul $P_3 = \begin{Vmatrix} 2 \\ 4 \end{Vmatrix}$ care determină astfel un sistem liniar independent banal.

Scanned with OKEN Scanner

În schimb, vectorul:

$$X_4 = \begin{Vmatrix} 3 \\ 0 \\ 1 \\ 0 \\ 0 \end{Vmatrix}$$ nu este soluție de bază.

Deși el verifică (1a) și (2a), fiind deci soluție posibilă, vectorii

$$P_1 = \begin{Vmatrix} 1 \\ 2 \end{Vmatrix} \text{ și } P_3 = \begin{Vmatrix} 2 \\ 4 \end{Vmatrix}$$

corespunzători componentelor nenule, formează un sistem liniar dependent, fapt pus în evidență de egalitatea

$$\begin{vmatrix} 1 & 2 \\ 2 & 4 \end{vmatrix} = 0.$$

b)
$$x_1 + 2x_2 = 3.$$

(1_b)
$$-x_2 + x_3 = 4,$$
$$3x_2 + x_4 = 2;$$

(2_b) $x_2 \geqslant 0, \quad x_1 \geqslant 0, \quad x_3 \geqslant 0, \quad x_4 \geqslant 0.$

Vectorul $X = \begin{Vmatrix} 3 \\ 0 \\ 4 \\ 2 \end{Vmatrix}$ este o soluție de bază nedegenerată.

Se remarcă că vectorii $P_1 = \begin{Vmatrix} 1 \\ 0 \\ 0 \end{Vmatrix}$, $P_3 = \begin{Vmatrix} 0 \\ 1 \\ 0 \end{Vmatrix}$, $P_4 = \begin{Vmatrix} 0 \\ 0 \\ 1 \end{Vmatrix}$

formează baza unității în spațiul vectorial R^3.

Să presupunem că X este o soluție de bază care are r componente $(r \leqslant m)$, $\bar{x}_{i_1}, \bar{x}_{i_2}, \ldots, \bar{x}_{i_r}$ nenule, iar restul sînt 0. Se poate arăta că sistemul liniar independent $P_{i_1}, P_{i_2}, \ldots, P_{i_r}$, poate fi completat pînă la o bază, prin adăogarea a încă $m - r$ vectori $P_{i_{r+1}}, P_{i_{r+2}}, \ldots, P_{i_m}$, dintre cei rămași.

Scanned with OKEN Scanner

În această situaţie condiţia (1') se poate scrie

$$\tilde{x}_{i_1} P_{i_1} + \tilde{x}_{i_2} P_{i_2} + \ldots + \tilde{x}_{i_m} P_{i_m} = P_0, \tag{4}$$

ceea ce constituie reprezentarea vectorului P_0 în baza $P_{i1}, P_{i2}, \ldots, P_{i_m}$.

Componentele $\tilde{x}_{i_1}, \tilde{x}_{i_2}, \ldots, x_{i_m}$ corespunzătoare vectorilor bazei (chiar şi cele nule) se vor numi *componente de bază* sau *bazice*, în timp ce restul componentelor lui $\overline{X}$ se vor numi *nebazice*.

Din cele relatate anterior se poate deduce modul de obţinere a soluţiilor de bază.

Fără a restrînge generalitatea, putem presupune că primele m componente ale soluţiei $\overline{X}$ sînt bazice. Aceasta implică faptul că vectorii $P_1, P_2, \ldots, P_m$ formează o bază în spaţiul restricţiilor.

Matricea A a coeficienţilor sistemului (1) se poate partiţiona, după cum urmează:

$$A = \left\| \begin{array}{ccc|ccc} a_{11} a_{12} & \ldots & a_{1m} & a_{1\,m+1} a_{1\,m+2} & \ldots & a_{1n} \\ a_{21} a_{22} & \ldots & a_{2m} & a_{2\,m+1} a_{2\,m+2} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{m1} a_{m2} & \ldots & a_{mm} & a_{m\,m+1} a_{m\,m+2} & \ldots & a_{mn} \end{array} \right\|.$$

Notînd cu B matricea formată din elementele primelor m coloane şi cu F matricea formată din elementele celorlalte $n - m$ coloane, A se scrie

$$A = ||B \mid F||. \tag{5}$$

Punînd în evidenţă vectorul-coloană $\overline{X}_B$ format din componentele bazice ale lui $\overline{X}$ şi vectorul-coloană $\overline{X}_F$ format din componentele nebazice, $\overline{X}$ va fi

$$\overline{X} = \left\| \begin{array}{c} \overline{X}_B \\ \overline{X}_F \end{array} \right\|.$$

Atunci egalitatea $A\overline{X} = b$, devine

$$B\overline{X}_B + F\overline{X}_F = b,$$

Scanned with OKEN Scanner

și cum $X_F = 0$,

$$B\bar{X}_B = b.$$

Din faptul că vectorii $P_1, P_2, \ldots, P_m$ sînt liniar independenți rezultă că matricea B este nesingulară și ca atare admite o inversă B^{-1}. Atunci

$$\bar{X}_B = B^{-1}b \tag{6}$$

relație potrivit căreia se obțin componentele bazice ale soluției de bază $\bar{X}$.

Observăm că acestea sînt soluțiile unui sistem de ecuații liniare, pătrat, cu m ecuații și m necunoscute, a cărui matrice nesingulară, este formată din vectorii unei baze din spațiul restricțiilor.

Cu alte cuvinte, pentru obținerea unei soluții de bază a problemei (S) se procedează astfel:

— se rezolvă sistemul pătrat de ecuații, obținut prin anularea celor $n - m$ variabile arbitrare din sistemul de ecuații liniare (1);

— dacă soluția unic determinată a sistemului este nenegativă, ea se poate completa pînă la o soluție de bază a problemei de programare, atribuind valoarea 0 celorlalte $n - m$ componente.

Evident, numărul soluțiilor de bază ale problemei (S) poate fi cel mult C_n^m, număr ce corespunde tuturor combinațiilor de m vectori ce se pot alcătui din vectorii $P_1, P_2, \ldots, P_n$.

În ipoteza făcută inițial asupra rangului matricei A este clar că programul liniar considerat admite soluții de bază.

Un rezultat privind existența soluțiilor de bază, poate fi formulat însă și înlăturînd ipoteza restrictivă. Se poate demonstra:

Teorema 2. *Dacă pentru un program liniar* $\mathfrak{X} \neq \emptyset$, *atunci există cel puțin o soluție de bază.*

După cum vom vedea ulterior soluțiile de bază se bucură de un interes special, în tehnica de rezolvare a programelor liniare.

Acest lucru va fi motivat de poziția lor specifică față de mulțimea soluțiilor posibile.

Dăm în continuare o teoremă deosebit de importantă, care are și o ilustrare geometrică foarte sugestivă.

Scanned with OKEN Scanner

Teorema 3. *Orice soluție de bază* $\overline{X}$ *a programului liniar constituie un punct extrem al mulțimii* $\mathfrak{X}$ *a soluțiilor posibile și reciproc, orice punct extrem al lui* $\mathfrak{X}$ *este o soluție de bază.*

Într-adevăr, să presupunem, prin absurd, că soluția de bază $\overline{X}$ nu ar fi punct extrem al lui $\mathfrak{X}$.

Atunci ar exista două soluții posibile diferite X^1, X^2 și un număr real λ, $0 < \lambda < 1$, astfel ca

$$\overline{X} = \lambda X^1 + (1 - \lambda) X^2. \tag{7}$$

Fără a restrînge generalitatea putem presupune că ultimele $n - r$, $(r \leqslant m)$ componente ale lui $\overline{X}$ sînt nule, și, corespunzător, vectorii $P_1, P_2, \ldots, P_r$ sînt liniar independenți.

Atunci din (7), rezultă imediat

$$x_i^1 + (1 - \lambda) x_i^2 = 0, \quad \text{pentru } i = r+1, \ldots, n;$$

or, cum x_i^1 și x_i^2 sînt nenegative, fiind componente ale soluțiilor posibile X^1 și X^2 și cum $0 < \lambda < 1$, rezultă de aici

$$x_i^1 = x_i^2 = 0, \text{ pentru } i = r + 1, \ldots, n. \tag{8}$$

De aici, ținînd seama de faptul că X^1 și X^2 satisfac egalitatea (1), rezultă

$$P_0 = x_1^1 P_1 + x_2^1 P_2 + \ldots + x_r^1 P_r = x_1^2 P_1 + x_2^2 P_2 + \ldots + x_r^2 P_r$$

Or, dată fiind independența vectorilor $P_1, P_2, \ldots, P_r$, ultima egalitate este posibilă doar dacă $x_i^1 = x_i^2$, pentru toți $i = 1, 2, \ldots, r$. ceea ce împreună cu (6), nu înseamnă altceva decît

$$X^1 = X^2$$

demonstrînd astfel prima parte a teoremei.

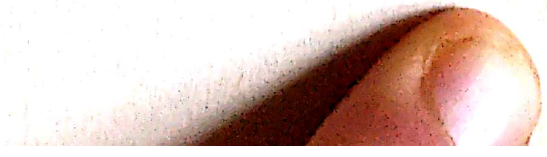
Pentru a doua parte a teoremei, fie X un punct extrem al mulțimii convexe $\mathfrak{X}$.

Presupunem că numai primele s componente ale sale sînt nenule (de fapt strict pozitive). Vom arăta că vectorii P_i corespunzători, sînt liniar independenți.

În caz contrar fie

$$\alpha_1 P_1 + \alpha_2 P_2 + \ldots + \alpha_s P_s = 0 \tag{9}$$

o combinație liniară nulă cu coeficienți nu toți nuli.



Scanned with OKEN Scanner

Cum X este soluție posibilă putem scrie:

$$x_1P_1 + x_2P_2 + \ldots + x_sP_s = P_0. \tag{10}$$

Scăzînd și adunînd din (10) relația (9) înmulțită cu un coeficient β obținem:

$$(x_1 - \beta\alpha_1)P_1 + (x_2 - \beta\alpha_2)P_2 + \ldots + (x_s - \beta\alpha_s)P_s = P_0,$$

$$(x_1 + \beta\alpha_1)P_1 + (x_2 + \beta\alpha_2)P_2 + \ldots + (x_s + \beta\alpha_s)P_s = P_0.$$

Dar, putem să-l alegem pe β suficient de mic, astfel încît $(x_i - \beta\alpha_i)$ și $(x_i + \beta\alpha_i)$, să fie nenegativi pentru toți $i = 1, 2, \ldots, s$, așa că ultimele două egalități definesc două soluții posibile ale programului.

Notînd X^1, X^2, vectorii de componente

$x_i^1 = (x_i - \beta\alpha_i)$, pentru $i = 1,2, \ldots,s$, $x^1 = 0$, pentru $i = s + 1, \ldots, n$ respectiv

$x_i^2 = (x_i + \beta\alpha_i)$, pentru $i = 1, 2, \ldots, s$, $x_i^2 = 0$, pentru $i = s + 1, \ldots, n$ este ușor de verificat că:

$$X = \frac{1}{2}X^1 + \frac{1}{2}X^2$$

ceea ce ar contrazice faptul că X este punct extrem al mulțimii $\mathfrak{X}$.

Deci vectorii P_i corespunzători componentelor nenule ale lui X sînt liniar independenți. De aici rezultă imediat $s \leqslant m$, deoarece numărul vectorilor liniar independenți este limitat de m.

În concluzie X este soluție de bază.

Vom defini, acum o noțiune centrală a teoriei programării liniare, *soluția optimă*.

Definiție. *Numim soluție optimă a programului liniar dat, o soluție posibilă X^* care satisface și cerința de optim* (3).

Cum scopul propus în rezolvarea problemelor de programare liniară este tocmai determinarea soluțiilor optime, le vom putea numi pe acestea, simplu, *soluții*.

Importanța soluțiilor de bază discutate anterior, va ieși și mai bine în evidență, din următorul rezultat:

Scanned with OKEN Scanner

Teorema 4. *Dacă programul liniar* (1), (2), (3), *are soluţie finită atunci există un punct extrem al mulţimii soluţiilor posibile care să constituie soluţie optimă.*

Pentru demonstrarea teoremei, este suficient să ţinem seama de faptul că orice punct X^* al mulţimii convexe $\mathfrak{X}$ se poate reprezenta ca o combinaţie liniară convexă a punctelor sale extreme.

Dacă X^* este soluţie optimă, atunci el va realiza maximul funcţiei obiectiv f. Fie M valoarea maximă a acesteia, presupusă finită.

Avem

$$M = f(X^*) = f\left(\sum_{i=1}^{l} \lambda_i X^i\right), \tag{11}$$

cu X^i puncte extreme ale lui $\mathfrak{X}$ şi $\sum_{i=1}^{l} \lambda_i = 1$.

Din liniaritatea funcţiei f, rezultă

$$f\left(\sum_{i=1}^{l} \lambda_i X^i\right) = \sum_{i=1}^{l} \lambda_i f(X^i). \tag{12}$$

Fie acum $f(X^j) = \max\limits_{1 \leqslant i \leqslant l} \{f(X^i)\}$.

Atunci din (11) şi (12) putem scrie

$$M \leqslant f(X^j) \sum_{j=1}^{l} \lambda_i = f(X^j).$$

Cum inegalitatea nu poate avea loc, rezultă

$$f(X^j) = M,$$

ceea ce nu înseamnă altceva decît că punctul extrem X^j este soluţie optimă a problemei.

Cum teorema 3 stabileşte egalitatea dintre mulţimea soluţiilor de bază şi mulţimea punctelor extreme ale lui $\mathfrak{X}$, din ultima teoremă rezultă clar că, dacă programul are soluţie finită, cel puţin una dintre acestea va fi atinsă pentru o soluţie de bază.

Următoarea teoremă dă o proprietate geometrică a mulţimii soluţiilor optime.

Scanned with OKEN Scanner

Teorema 5. *Mulțimea soluțiilor optime ale unui program liniar este convexă.*

Cazurile banale în care mulțimea soluțiilor optime este vidă sau se reduce la un singur punct, verifică evident teorema.

Să presupunem că programul admite două soluții diferite X^1 și X^2, și fie λ, $0 \leqslant \lambda \leqslant 1$.

Din teorema 1 rezultă că $X = \lambda X^1 + (1 - \lambda)X^2$ este soluție posibilă.

Fie $M = \max f(X)$, pentru toți $X \in \mathfrak{X}$.

Atunci $f(X^1) = f(X^2) = M$.

Pe de altă parte

$$f(X) = \lambda f(X^1) + (1 - \lambda)f(X^2) = M$$

ceea ce probează, optimalitatea lui X.

Ca o consecință a acestei teoreme putem afirma că dacă un program liniar își atinge optimul în mai multe puncte diferite atunci el și-l va atinge, de asemenea, în orice alt punct care se poate exprima printr-o combinație liniară convexă a acestor puncte.

§ 5. METODA SIMPLEX DE REZOLVARE A PROGRAMELOR LINIARE

5.1. Principiile metodei

Din principalele teoreme ale paragrafului precedent, putem trage concluzia că pentru rezolvarea unui program liniar este necesar și suficient să cercetăm toate soluțiile de bază, pentru a le descoperi pe acelea care optimizează funcția obiectiv.

O inspectare completă a mulțimii soluțiilor de bază presupune pe lîngă volumul mare de calcule necesare pentru determinarea ei și o serie de alte dezavantaje. De exemplu în cazul în care problema are soluție infinită, metoda nu ar putea indica acest lucru.

Metoda simplex prezintă avantajul, că pe lîngă un volum relativ mic de calcule, poate da răspunsuri precise și concludente tuturor situațiilor pe care le poate prezenta, sub raportul rezolvabilității, programul liniar.

Scanned with OKEN Scanner

În esență, metoda constă într-o succesiune de cicluri, fiecare din ele dînd o soluție de bază, de o asemenea manieră încît valorile funcției obiectiv se îmbunătățesc continuu, pînă la atingerea optimului (dacă există).

Să considerăm programul adus la forma standard:

$$AX = b; \tag{1}$$

$$X \geqslant 0; \tag{2}$$

$$\text{Max}\{f(X) = CX\}. \tag{3}$$

Cu notațiile introduse în paragraful 3, să presupunem că vectorii coloană ai matricei A, P_{i_1}, P_{i_2}, ..., P_{i_m} formează o bază în spațiul restricțiilor.

Fie $Z_j = (z_{j1}, z_{j2}, \ldots, z_{jm})$ exprimarea vectorului P_j, $(j = 1,2,\ldots,n)$ în baza B specificată mai sus.

Fie R mulțimea indicilor j corespunzători vectorilor care nu fac parte din baza B și $\overline{R}$ mulțimea indicilor vectorilor bazici (așa că $R \cup \overline{R} = \{1,2,\ldots,n\}$)

Notăm cu C_B vectorul linie format din componentele lui C de ranguri $i_1, i_2, \ldots, i_m$ și cu z_j produsul $C_B Z_j$. (Aici, ca și în alte ocazii similare Z_j este considerat vector coloană).

Fie acum o soluție de bază $\overline{X}$, corespunzătoare bazei B (adică $\bar{x}_i = 0$, pentru $i \in R$).

Teorema 1 (testul de optimalitate). *Dacă $c_j - z_j \leqslant 0$, pentru toți $j = 1,2, \ldots, n$, atunci problema de programare liniară are optim finit și $\overline{X}$ este o soluție optimă.*

Într-adevăr, fie X o soluție posibilă oarecare a programului.

Putem deci scrie

$$P_0 = \sum_{j=1}^{n} x_j P_j \tag{4}$$

și

$$P_0 = \sum_{k=1}^{m} \bar{x}_{i_k} P_{i_k}. \tag{5}$$

Scanned with OKEN Scanner

De asemenea, deoarece B este bază, avem

$$P_j = \sum_{k=1}^{m} z_{jk} P_{i_k}.$$

Din (4), (5) și ultima egalitate,

$$\sum_{k=1}^{m} \bar{x}_{i_k} P_{i_k} = \sum_{j=1}^{n} x_j \sum_{j=1}^{m} z_{jk} P_{i_k}$$

de unde,

$$\bar{x}_{i_k} = \sum_{j=1}^{n} x_k z_{jk}.$$

Atunci

$$(6) \quad \begin{aligned} f(X) - f(\bar{X}) &= \sum_{j=1}^{n} c_j x_j - \sum_{k=1}^{m} c_{i_k} \bar{x}_{i_k} = \\ &= \sum_{j=1}^{n} c_j x_j - \sum_{k=1}^{m} c_{i_k} \sum_{j=1}^{n} x_j z_{ik} = \sum_{j=1}^{n} x_j (c_j - z_j) \end{aligned}$$

și deci, în baza ipotezei, deoarece $x_j \geqslant 0$, rezultă

$$f(X) \leqslant f(\bar{X}).$$

Observație. În problemele de minim, textul de optimalitate este $c_j - z_j \geqslant 0$ sau $z_j - c_j \leqslant 0$.

În cazul în care $\bar{X}$ nu este optim, metoda simplex prescrie alegerea unei alte soluții de bază îmbunătățite, adică un punct $\bar{\bar{X}}$ pentru care $f(\bar{\bar{X}}) \geqslant f(\bar{X})$. Particularitatea noii soluții selectate constă însă în faptul că va fi un punct extrem al mulțimii $\mathfrak{X}$, vecin cu cel reprezentat de soluția de bază $\bar{X}$.

Acest lucru se poate obține înlocuind unul din vectorii bazei B cu un vector nebazic P_j, $(j \in R)$, cu condiția ca noul sistem de vectori să rămînă liniar independent,

Vom arăta mecanismul de selecție utilizat în cadrul metodei simplex în scopul determinării vectorului P_j pentru ca noua soluție de bază rezultată să fie mai bună decît precedenta.

Scanned with OKEN Scanner

După cum am mai spus, în acest proces trebuie să urmărim, în primul rînd, ca noua soluţie să fie de asemenea una de bază.

Fie dar P_i vectorul bazei B care se înlocuieşte cu P_j, $(i \in \overline{R}, j \in R)$ şi fie $\overline{R}'$ mulţimea indicilor ce se obţine din $\overline{R}$ prin înlocuirea lui i cu j. Să notăm cu y_k, $(k \in \overline{R}')$ componentele bazice ale noii soluţii $\overline{\overline{X}}$.

Avem

$$P_0 = \sum_{k \in \overline{R}} \bar{x}_k P_k = \sum_{k \in \overline{R}'} y_k P_k. \tag{7}$$

Dar

$$\sum_{k \in \overline{R}'} y_k P_k = \sum_{\substack{k \in \overline{R} \\ k \neq i}} y_k P_k + y_j P_j = \sum_{\substack{k \in \overline{R} \\ k \neq i}} y_k P_k + y_j \sum_{k \in \overline{R}} z_{jk} P_k =$$

$$= \sum_{\substack{k \in \overline{R} \\ k \neq i}} (y_k + y_j z_{jk}) P_k + y_j z_{ji} P_i$$

aceasta, deoarece P_j se poate reprezenta ca o combinaţie liniară a vectorilor bazei B.

Ţinînd seama de (7) putem scrie

$$\begin{aligned} \bar{x}_i &= y_j z_{ji}, \\ \bar{x}_k &= y_k + y_j z_{jk}, \text{ pentru } k \in \overline{R}, k \neq i \end{aligned} \tag{8}$$

De aici

$$\begin{aligned} y_j &= \frac{\bar{x}_i}{z_{ji}}, \\ y_k &= \bar{x}_k - \frac{\bar{x}_i}{z_{ji}} z_{jk}, \text{ pentru } k \in \overline{R},\ k \neq i. \end{aligned} \tag{9}$$

Deoarece am presupus însă, că $\overline{X}$ este o soluţie de bază, trebuie ca toate componentele sale să fie nenule. Atunci din (9) rezultă

$$\frac{\bar{x}_i}{z_{ji}} \geqslant 0, \text{ deci } z_{ji} > 0. \tag{10}$$

$$\bar{x}_k - \frac{\bar{x}_i}{z_{ji}} z_{jk} \geqslant 0, \quad \text{deci } \frac{\bar{x}_i}{z_{ji}} \leqslant \frac{\bar{x}_k}{z_{jk}}, \text{ pentru } k \in \overline{R},\ k \neq i.$$

Scanned with OKEN Scanner

Relaţiile (10) constituie condiţiile necesare pe care trebuie să le îndeplinească vectorul P_i, pentru a putea fi înlocuit, în formarea unei noi soluţii de bază, de către vectorul P_j, presupus cunoscut.

În continuare vom determina criteriile de alegere a acestuia din urmă, punînd condiţia $f(\overline{\overline{X}}) > f(\overline{X})$.

Ţinînd seama de (9), această condiţie se transcrie

$$(11) \qquad 0 < \sum_{k \in \overline{R}'} c_k y_k - \sum_{k \in \overline{R}} c_k \bar{x}_k =$$

$$= c_j y_j + \sum_{\substack{k \in \overline{R} \\ k \neq i}} c_k \left(\bar{x}_k - \frac{\bar{x}_i}{z_{ji}} z_{jk} \right) - c_i \bar{x}_i - \sum_{\substack{k \in \overline{R} \\ k \neq i}} c_k \bar{x}_k =$$

$$= c_j \frac{\bar{x}_i}{z_{ji}} - \frac{z_i}{z_{ji}} \left(\sum_{\substack{k \in \overline{R} \\ k \neq i}} c_k z_{jk} + c_i z_{ji} \right) = \frac{\bar{x}_i}{z_{ji}} (c_j - z_j).$$

Cum, din (10), trebuie ca $z_{ji} > 0$, inegalitatea (11) stabileşte

$$(12) \qquad c_j - z_j > 0.$$

În concluzie putem formula

Teorema 2 (criteriul de îmbunătăţire a soluţiei). *Dacă există o pereche de indici i şi j ($i \in \overline{R}$, $j \in R$) astfel ca relaţiile* (12) *şi* (10) *să fie satisfăcute, atunci egalităţile* (9) *permit obţinerea unei soluţii de bază îmbunătăţite.*

Exemplul 4 al paragrafului 2 sugerează posibilitatea ca un program liniar să aibe optim infinit.

Această situaţie poate fi depistată precis în cadrul metodei simplex, dacă ţinem seama de următorul rezultat:

Teorema 3 (recunoaşterea infinitudinii soluţiei). *Dacă pentru un indice j, pentru care $c_j - z_j > 0$, toate componentele $z_{jk} \leqslant 0$, ($k \in \overline{R}$), atunci programul are optim infinit.*

Într-adevăr, putem încerca să construim o soluţie posibilă a problemei cu $m + 1$ componente nenule asociate vectorilor bazei B şi vectorului P_j.

Atunci din egalitatea

$$P_0 = \sum_{k \in \overline{R}} \bar{x}_k P_k = \sum_{k \in \overline{R}} y_k P_k + y_j P_j$$

Scanned with OKEN Scanner

deducem

$$\bar{x}_k = y_k + y_j z_{jk}, \text{ pentru } k \in \bar{R} \text{ sau } k = j.$$

Dacă $z_{jk} \leqslant 0$, rezultă de aici că valoarea componentei y_j, a noii soluţii, poate fi oricît de mare, $\bar{x}_k$, rămînînd în continuare nenegative.

De aici şi din ipoteza $c_j - z_j > 0$, rezultă şi nemărginirea superioară a valorilor funcţiei obiectiv.

5.2. Descrierea algoritmului

Algoritmul simplex constă dintr-o succesiune de iterate, care îmbunătăţesc treptat o soluţie iniţială de bază.

Presupunem că dispunem de o soluţie de bază X°. Fără a restrînge generalitatea, putem presupune că vectorii bazici corespunzători sînt $P_1, P_2, \ldots, P_m$. Fie z_{jk} ($j = 1,2,\ldots, n$; $k = 1,2,\ldots, m$) componentele vectorilor P_j în baza B determinată de vectorii specificaţi mai sus. Se calculează diferenţele $c_j - z_j$ şi se completează tabelul 1.

Tabelul 1

C_B	$c_j \rightarrow$		c_1	...	c_i	...	c_m	c_{m+1}	...	c_j	...	c_n
↓	Variabile bazice	P_0	P_1	...	P_i	...	P_m	P_{m+1}	...	P_j	...	P_n
c_1	x_1	x_1^0	1	...	0	...	0	$z_{m+1,1}$	...	z_{j_1}	...	z_{n1}
...	...	...	...	...	...	...	...	...	...	...	...	...
c_i	x_i	x_i^0	0	...	1	...	0	z_{m+1i}	...	z_{ji}	...	z_{ni}
...	...	...	...	...	...	...	...	...	...	...	...	...
c_m	x_m	x_m^0	0	...	0	...	1	$z_{m+1,n}$	...	z_{jm}	...	z_{nm}
	$c_j - z_j$	$z_0 = C_B P_0$	0	...	0	...	0	$c_{m+1} - z_{m+1}$	...	$c_j - z_j$	...	$c_n - z_n$

Se pot face următoarele observaţii:

1) Componentele vectorilor bazici sînt date de vectorii unitari din spaţiul restricţiilor.

Scanned with OKEN Scanner

2) Diferențele $c_j - z_j$ sînt nule pentru vectorii bazici.

3) $z_0 = C_B P_0$ reprezintă valoarea funcției obiectiv corespunzătoare soluției de bază.

După completarea tabelului se testează optimalitatea soluției conform teoremei 1. Dacă răspunsul este afirmativ problema poate fi considerată rezolvată deoarece s-a aflat o soluție optimă (în general, nu se urmărește obținerea tuturor soluțiilor optime, ci numai a uneia din ele.)

În caz contrar, dacă există j cu $c_j - z_j > 0$, (conform observației făcute anterior, j nu poate fi decît în mulțimea indicilor nebazici), soluția inițială nu este optimă.

Există două posibilități.

Dacă corespunzător unui indice j, pentru care diferența $c_j - z_j$ este strict pozitivă, toți $z_{ji} \leqslant 0$, $(i = 1,2,\ldots,m)$, atunci programul are optim infinit (teorema 3), și ca atare determinarea sa devine neinteresantă.

Altfel, condițiile (12) și (10) sînt satisfăcute și deci soluția se poate îmbunătăți.

În acest caz se inițiază prima iterată a algoritmului. Aceasta revine la determinarea vectorilor P_j, P_i, care intră, respectiv, ies din bază.

De regulă P_j se determină conform următoarelor criterii:

— diferența $c_j - z_j$ corespunzătoare să aibă valoarea strict pozitivă maximă;

— cel puțin o componentă z_{jk} să fie strict pozitivă, (condiția asupra maximului valorii lui $c_j - z_j$ nu este obligatorie dar s-a constatat, că în majoritatea cazurilor are ca efect micșorarea numărului de iterate necesare).

Vectorul P_i se determină conform inegalităților (10) corespunzător indicelui pentru care

$$\frac{x_i^0}{z_{ji}} = \min \frac{x_k^0}{z_{jk}},$$

minimul fiind luat pentru rapoartele în care $z_{jk} > 0$.

Odată cei doi vectori stabiliți, vechea variabilă bazică x_i se înlocuiește cu x_j, și se recalculează toate elementele tabelului precedent.

Scanned with OKEN Scanner

Noile componente ale lui P_0, x_k^1 se deduc din formulele (9)

$$x_j^1 = \frac{x_i}{z_{ji}},$$

$$x_k^1 = x_k^0 - \frac{z_i^0}{z_{ji}} z_{jk}, \quad k \neq i. \tag{9'}$$

Noile componente z_{sk}^1 ale vectorilor P_s se obţin, în urma schimbării operate în bază, conform egalităţilor

$$z_{sj}^1 = \frac{z_{si}}{z_{ji}},$$

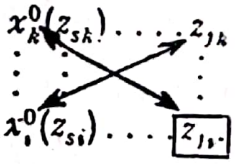
$$z_{sk}^1 = z_{sk} - \frac{z_{si}}{z_{ji}} z_{jk}, \quad k \neq i \tag{9''}$$

(egalităţile se obţin, printr-un raţionament analog cu cel care ne-a condus la relaţiile (9), plecînd de la reprezentarea vectorului P_s în cele două baze).

Ca efect al formulelor (9'), (9''), mecanismul de calcul este următorul:

— toate elementele situate pe linia vechii variabile bazice x_i se împart cu z_{ji} (pe care îl vom numi *pivot*);

— restul elementelor tabloului se transformă după regula dreptunghiului



Adică, noul element al tabelului se calculează cu ajutorul celor patru elemente menţionate în schemă astfel: diferenţa dintre produsul celor două elemente ale diagonalei dreptunghiului care conţine pivotul şi produsul elementelor de pe cealaltă diagonală, se imparte la elementul pivot.

Prin recalcularea elementelor se obţine tabelul 2.

După completarea tabelului, cu diferenţele $c_k - z_k$ calculate, se testează optimalitatea noii soluţii. Dacă noua soluţie este optimă, algoritmul se opreşte. În caz contrar, se pot pune două alternative discutate anterior. Dacă soluţia poate fi îmbunătăţită se începe o nouă iterată.

Scanned with OKEN Scanner

Tabelul 2

C_B	$c_j \rightarrow$		c_1	...	c_i	...	c_m	c_{m+1}	...	c_j	...	c_n
↓	Variabile bazice	P_0	P_1	...	P_i	...	P_m	P_{m+1}	...	P_j	...	P_n
c_1	x_1	x_1^1	1	...	z_i^1	...	0	$z_{m+1,1}$	...	0	...	z_{n1}
...	...	...	...	...	...	...	...	...	...	...	...	...
c_j	x_j	x_j^1	0	...	z_{ij}^1	...	0	$z_{m+1,j}$	...	1	...	z_{nj}
...	...	...	...	...	...	...	...	...	...	...	...	...
c_m	x_m	x_m^1	0	...	z_{im}^1	...	1	$z_{m+1,m}$	...	0	...	z_{nm}
	$c_j - z_j$	$z_0 = C_B P_0$	0	...	$c_i - z_i$	...	0	$c_{m+1} - z_{m+1}$	...	0	...	$c_n - z_n$

În cazul cînd testul de optimalitate dă răspuns afirmativ, ultimul tabel simplex conţine soluţia programului. Componentele lui P_0 reprezintă chiar componentele bazice ale soluţiei optime (celelalte sînt nule), iar z_0 reprezintă valoarea optimă a funcţiei obiectiv.

Exemplul 1. Să se rezolve programul liniar

$$\begin{aligned} &2x_1 + x_2 + x_3 - x_4 + x_6 = 2, \\ &-x_1 + 2x_2 - 2x_3 - 2x_4 + x_7 = 3, \\ &3x_1 - x_2 + x_3 + x_5 = 9 \\ &x_i \geqslant 0, \qquad i = 1,2\ldots,7. \end{aligned}$$

$$\min \left(3x_1 - 7x_2 + \frac{1}{2} x_3 + 5x_4 + 3x_5\right)$$

Cerinţa de minim poate fi înlocuită prin cea de maxim, înmulţind cu -1, funcţia obiectiv.

$$\max \left(-3x_1 + 7x_2 - \frac{1}{2} x_3 - 5x_4 - 3x_5\right).$$

Matricea A a coeficienţilor sistemului de ecuaţii este

$$\left\| \begin{array}{rrrrrrr} 2 & 1 & 1 & -1 & 0 & 1 & 0 \\ -1 & 2 & -2 & -2 & 0 & 0 & 1 \\ 3 & -1 & 1 & 0 & 1 & 0 & 0 \end{array} \right\|.$$

Soluţia optimă finită a programului se obţine după trei iteraţii simplex, cuprinse în tabelul 3.

Explicaţii.

Pentru obţinerea soluţiei de bază iniţiale, am făcut observaţia că vectorii P_6, P_7, P_5 formează o bază, fiind vectori unitari în R^3.

Ca atare componentele celorlalţi vectori în această bază sînt chiar cele care apar în coloanele matricii A. În acest caz $P_0 = b$, şi deoarece termenii liberi sînt toţi nenegativi, componentele lui b sînt componente bazice pentru o primă soluţie de bază.

Cu aceasta s-a completat prima parte din tabelul 3.

Deoarece există diferenţe $c_j - z_j$ strict pozitive, soluţia nu este optimă.

Imbunătăţirea este posibilă, deoarece vectorul P_1, (pentru care diferenţa $c_1 - z_1 = 6 > 0$), are componente strict pozitive. Ca atare vectorul P_1 intră în bază.

Pentru determinarea vectorului care părăseşte baza, am căutat minimul rapoartelor pozitive $\frac{x_k^0}{z_{1k}}$.

Cum valorile acestora sînt $\frac{2}{2}$ şi $\frac{9}{3}$, minimul este atins corespunzător variabilei x_6. Deci P_6 părăseşte baza, $z_{16} = 2$ este element pivot.

Rezultatele primei iterate apar în partea a doua a tabelului 3.

Noua soluţie de bază are componentele bazice

$$x_1 = 1, \quad x_7 = 4, \quad x_5 = 6.$$

Deoarece $c_2 - z_2 = 1 > 0$, şi P_2 are componente strict pozitive, soluţia poate fi îmbunătăţită.

$$\min\left\{\frac{1}{1/2}, \frac{4}{5/2}\right\} = \frac{4}{5/2},$$

deci P_7 iese din bază, intrînd P_2.

Scanned with OKEN Scanner

Tabelul 3

C_B	$c_j \rightarrow$		-3	7	$-1/2$	-5	-3	0	0
$\downarrow$	Variabile bazice	P_0	P_1 $\downarrow$	P_2	P_3	P_4	P_5	P_6	P_7
0	$\leftarrow x_6$	2	$\boxed{2}$	1	1	-1	0	1	0
0	x_7	3	-1	2	-2	-2	0	0	1
-3	x_5	9	3	-1	1	0	1	0	0
	$c_j - z_j$	$z_0 = -27$	6	4	5/2	-5	0	0	0
-3	x_1	1	1	1/2	1/2	$-1/2$	0	1/2	0
0	$\leftarrow x_7$	4	0	$\boxed{5/2}$	$-3/2$	$-5/2$	0	1/2	1
-3	x_5	6	0	$-5/2$	$-1/2$	3/2	1	$-3/2$	0
	$c_j - z_j$	$z_0 = -21$	0	1	$-1/2$	-2	0	-3	0.
-3	$\leftarrow x_1$	1/5	1	0	$\boxed{4/5}$	0	0	2/5	$-1/5$
7	x_2	8/5	0	1	$-3/5$	-1	0	1/5	2/5
-3	x_5	10	0	0	-2	-1	1	-1	1
	$c_j - z_j$	$z_0 = -97/5$	0	0	1/10	-1	0	$-16/5$	$-2/5$
$-1/2$	x_3	1/4	5/4	0	1	0	0	1/2	$-1/4$
7	x_2	7/4	3/4	1	0	-1	0	1/2	1/4
-3	x_5	21/2	5/2	0	0	-1	1	0	1/2
	$c_j - z_j$	$-155/8$	$-1/8$	0	0	-1	0	$-13/4$	$-3/8$

Elementele calculate în a doua iterată apar în parta a treia a tabelului 3. Noua soluţie de bază are componentele nenule

$$x_1 = 1/5, \quad x_2 = 8/5, \quad x_5 = 10.$$

Scanned with OKEN Scanner

Aici $c_3 - z_3 = 1/10 > 0$, și $z_{31} = 4/5 > 0$, deci intră P_3, iese P_1, pivot fiind $z_{31} = 4/5$.

În urma celei de a treia iterate, rezultă partea a patra a tabelului 3.

Aici toate diferențele $c_j - z_j \leqslant 0$, deci, algoritmul ia sfîrșit, cu obținerea optimului finit.

Soluția optimă

$$x_2^* = 7/4,\ x_3^* = 1/4,\ x_{5!}^* = 21/2,\ x_1^* = x_4^* = x_6^* = x_7^* = 0.$$

Valoarea optimă a funcției în problema de maxim este $-155/8$.

Valoarea optimă a funcției obiectiv în problema inițială de minim este 155/8 (deoarece $\min f(X) = -\max(-f(X))$.

Exemplul 2. Să se rezolve programul liniar

$$\begin{aligned}
&2x_1 - x_2 + x_3 - 8x_4 \qquad + x_6 \qquad = 20,\\
&x_1 - x_2 + 2x_3 - 3x_4 + x_5 \qquad = 8,\\
&x_1 - 3x_2 + x_3 \qquad + x_7 = 4,\\
&x_i \geqslant 0,\ i = 1,2,\ldots,7;\\
&\max\ (-2x_1 + x_2 - 4x_3 + 10x_4 - 2x_5 - 3x_6 + x_7)
\end{aligned}$$

Matricea coeficienților este

$$A = \begin{Vmatrix} 2 & -1 & 1 & -8 & 0 & 1 & 0 \\ 1 & -1 & 2 & -3 & 1 & 0 & 0 \\ 1 & -3 & 1 & 0 & 0 & 0 & 1 \end{Vmatrix}$$

Sîntem conduși la succesiunea de iterații simplex date în tabelul 4.

Ca și în exemplul precedent, prima parte a tabelului 4 se completează imediat, făcînd observația că vectorii primei baze sînt vectorii unitari P_6, P_5, P_7.

Cum $c_1 - z_1 = 5 > 0$, și P_1 are componente strict pozitive, soluția inițială poate fi îmbunătățită.

$$\min\left\{\frac{20}{2},\ \frac{8}{1},\ \frac{4}{1}\right\} = \frac{4}{1},$$

deci, P_7 iese din bază, intrînd P_1; elementul pivot este $z_{17} = 1$.

Scanned with OKEN Scanner

Tabelul 4

C_B	$c_j \rightarrow$		-2	1	-4	10	-2	-3	1
$\downarrow$	Variabile bazice	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
-3	x_6	20	2	-1	1	-8	0	1	0
-2	x_5	8	1	-1	2	-3	1	0	0
1	$\leftarrow x_7$	4	$\boxed{1}$	-3	1	0	0	0	1
	$c_j - z_j$	$z_0 = -72$	5	-1	2	-20	0	0	0
-3	x_6	12	0	5	-1	-8	0	1	-2
-2	$\leftarrow x_5$	4	0	$\boxed{2}$	1	-3	1	0	-1
-2	x_1	4	1	-3	1	0	0	0	1
	$c_j - z_j$	$z_0 = -52$	0	14	-3	-20	0	0	-5
-3	x_6	2	0	0	$-7/2$	$-1/2$	$-5/2$	1	$1/2$
1	x_2	2	0	1	$1/2$	$-3/2$	$1/2$	0	$-1/2$
-2	x_1	10	1	0	$5/2$	$-9/2$	$3/2$	0	$-1/2$
	$c_j - z_j$	$z_0 = -24$	0	0	-10	1	-7	0	2

În partea doua a tabelului 4 au fost trecute elementele calculate în prima iterată.

Algoritmul continuă deoarece

$c_2 - z_2 = 14 > 0$ și P_2 are componente strict pozitive

$$\min\left\{\frac{12}{5}, \frac{4}{2}\right\} = \frac{4}{2}.$$

Scanned with OKEN Scanner

Intră P_2 în locul lui P_5, pivotul fiind $z_{25} = 2$,
Am ajuns la partea a treia a tabelului 4.

Deoarece $c_4 - z_4 = 1 > 0$, dar P_4 nu are nici o componentă pozitivă în baza corespunzătoare, rezultă nemărginirea programului.

Descrierea algoritmului, s-a făcut în prezenta secțiune, plecînd de la o presupusă soluție de bază inițială. Desigur, determinarea unei soluții de bază nu constituie, principial, o chestiune delicată. În § 4 am văzut care este metoda generală de determinare a unor asemenea soluții.

Totuși, trebuie să remarcăm că metoda expusă acolo se dovedește laborioasă, comportînd, pe de o parte o discuție asupra compatibilității sistemului de restricții, iar pe de altă parte, pentru algoritmul simplex, necesitatea unui volum mare de calcule pentru reprezentarea tuturor vectorilor P în baza corespunzătoare.

Vom vedea în continuare calea indicată, în cadrul metodei simplex, pentru stabilirea unei soluții inițiale de bază.

5.3. Baza artificială. Metoda celor două faze

Pentru evitarea căutării primei soluții posibile necesare, în cazul cînd matricea A nu conține toți vectorii unitari, se consideră următoarea problemă extinsă:

$$(1^a) \qquad AX + EX^a = b,$$

$$(2^a) \qquad X \geqslant 0, \quad X^a \geqslant 0,$$

$$(3^a) \qquad \max\{CX - \alpha(X^a)'\}$$

unde, E este matricea unitate m-dimensională, X^a este un vector m-dimensional ale cărui componente le notăm cu x_k^a, $(k = n + 1, n + 2, \ldots, n + m)$, iar α un număr real arbitrar strict pozitiv. Se va presupune de asemenea $b \geqslant 0$.

Construcția problemei extinse se face de fapt, adăugînd în membrul stîng al fiecărei ecuații din sistemul (1) cîte o variabilă artificială pe care conform cu (2^a) o vom presupune nenegativă.

Scanned with OKEN Scanner

În același timp modificăm și funcția obiectiv scăzînd din forma liniară inițială noile variabile înmulțite cu o aceeași constantă pozitivă.

Este clar că orice soluție posibilă a problemei inițiale devine o soluție posibilă a programului extins pentru care valorile tuturor variabilelor artificiale sînt 0.

Reciproc, orice soluție posibilă a programului extins în care toate variabilele artificiale sînt nule, devine, prin înlăturarea acestora, o soluție posibilă a programului originar.

În concluzie, dacă problema inițială are soluții posibile acestea se vor putea obține numai din soluțiile posibile ale programului extins, de forma specificată mai sus.

Or, este ușor de remarcat că asemenea soluții realizează, în mod obligatoriu:

$$(4^a) \qquad \max(-\alpha(X^a)).$$

Aceasta deoarece din pozitivitatea lui α și din (2^a), expresia respectivă nu poate depăși, în nici un caz, valoarea 0, și această valoare maximă nu poate fi atinsă decît pentru $X^a = 0$.

Observațiile de mai sus au stat la baza metodei celor două faze, care, împreună cu algoritmul descris anterior completează metoda simplex.

Faza I.

În prima fază se rezolvă, folosind algoritmul simplex programul: (1^a), (2^a), (4^a), luînd drept bază inițială pe E (bază artificială, a vectorilor corespunzători variabilelor artificiale).

Cum, evident, situația optimului infinit nu poate apare aici, la sfîrșitul acestei faze ne putem găsi în una din următoarele alternative:

i) Valoarea optimă din (4^a) este strict negativă.

ii) Valoarea optimă din (4^a) este 0 și nici o variabilă artificială nu este bazică față de soluția optimă.

iii) Valoarea optimă din (4^a) este 0 și cel puțin o variabilă artificială este bazică.

În cazul i), conform observațiilor precedente programul inițial nu are soluții posibile fiind deci, nerezolvabil.

În cazul ii) dispunem de o soluție de bază a programului extins din care, prin înlăturarea componentelor artificiale nule

Scanned with OKEN Scanner

se obține o soluție de bază a programului inițial. Cu aceasta drept soluție inițială se începe faza II.

Dacă se realizează alternativa iii), soluția de bază conține și componente artificiale, care vor fi însă, obligatoriu, nule (deoarece, în caz contrar valoarea optimă (4^a), nu ar putea fi 0).

Or, în acest caz, dispunem de o soluție de bază pentru programul inițial, însă una degenerată (avînd mai puțin de m componente nenule). Se trece și în acest caz la faza II, avînd însă grije ca pe parcurs variabilele artificiale rămase în bază să nu ia valori diferite de 0.

Faza a II-a.

În situația ii) a doua fază constă în rezolvarea programului inițial prin algoritmul simplex, avînd drept soluție de bază inițială pe cea determinată la sfîrșitul primei faze. Pentru aceasta se îndepărtează din ultimul tabel simplex elementele corespunzătoare variabilelor artificiale și introducînd coeficienții funcției obiectiv $f(X) = CX$, se continuă șirul iteratelor simplex.

Dacă finalul primei faze se situează în cazul iii), faza II recomandă aplicarea metodei simplex unui program restrîns obținut astfel: se îndepărtează din programul extins toate variabilele nebazice pentru care în ultima iterată $c'_j - z_j < 0$, păstrîndu-se, în schimb, variabilele bazice (am notat cu c'_j coeficienții funcției obiectiv din (4^a)). De asemenea se revine la funcția obiectiv inițială.

Explicația acestui procedeu este imediată; din relația (6), rezultă că soluția optimă a programului rezolvat în prima fază, împreună cu orice altă soluție posibilă a acestuia, satisface egalitatea

$$(13) \qquad -\alpha X^a = \sum_{k=1}^{n} x(c'_j - z_j) + \sum_{j=n+1}^{m} x_j^a (c'_j - z_j),$$

($c'_j = 0$ sau $-\alpha$, după cum $j = 1, 2, \ldots, n$, sau $j = n+1, \ldots$ $\ldots, n+m$).

Dar cum soluția finală implică toți $c'_j - z_j \leqslant 0$, rezultă că pentru ca membrul stîng din (13) să fie nul este obligatoriu ca $x_j = 0$ pentru acei $c'_j - z_j$ strict negativi. Or, pentru rezolvarea programului inițial, iteratele simplex viitoare vor trebui să opereze numai cu acele soluții posibile provenite din soluții optime

Scanned with OKEN Scanner

ale programului pus în faza I, adică cu acelea pentru care $-\alpha X^a = 0$. În efect, toate variabilele x_j, cu $c'_j - z_j < 0$ la sfîrşitul primei faze vor fi menţinute în permanenţă nule şi ca atare, pot fi eliminate din program.

În concluzie pentru iniţierea fazei II în situaţia iii), se elimină din ultimul tabel simplex coloanele corespunzătoare diferenţelor $c'_j - z_j < 0$ şi se introduc coeficienţii funcţiei obiectiv $f(X) = CX$.

Înainte de exemplificarea metodei, expuse prin aplicarea ei unor programe numerice, să facem observaţie că nu este obligatorie introducerea a m variabile artificiale. Practic, dacă în matricea A există un număr oarecare de vectori unitari diferiţi, se adaugă numai variabilele artificiale corespunzătoare vectorilor unitari lipsă pînă la completarea matricei unitate.

Exemplul 3. Să se rezolve programul liniar

$$2x_1 + x_2 + x_3 - x_4 = 2,$$

$$x_1 - 2x_2 + 2x_3 + 2x_4 = -3,$$

$$3x_1 - x_2 + x_3 + x_5 = 9,$$

$$x_i \geqslant 0,\ i = 1, 2, \ldots, 5;$$

$$\max\ (-3x_1 + 7x_2 - 1/2x_3 - 5x_4 - 3x_5).$$

Deoarece $b_2 = -3 < 0$, se înmulţeşte ecuaţia a doua cu -1 obţinînd

$$-x_1 + 2x_2 - 2x_3 - 2x_4 = 3.$$

Matricea A a sistemului de ecuaţii este

$$A = \begin{Vmatrix} 2 & 1 & 1 & -1 & 0 \\ -1 & 2 & -2 & -2 & 0 \\ 3 & -1 & 1 & 0 & 1 \end{Vmatrix}.$$

Vectorul P_5 este unul din vectorii unitari din R^3, ca atare pentru aplicarea metodei celor două faze este necesară introducerea vectorilor unitari e_1, e_2 şi corespunzător lor, a variabilelor artificiale x_6^a şi x_7^a în prima, respectiv, a doua ecuaţie.

Scanned with OKEN Scanner

Faza I

Se rezolvă programul liniar

$$\begin{aligned} &2x_1 + x_2 + x_3 - x_4 + x_6^a = 2,\\ &-x_1 + 2x_2 - 2x_3 - 2x_4 + x_7^a = 3,\\ &3x_1 - x_2 + x_3 + x_5 = 9,\\ &x_i \geqslant 0,\ i = 1, 2, \ldots, 5,\ x_6^a \geqslant 0,\ x_7^a \geqslant 0.\\ &\max(-\alpha x_6^a - \alpha x^a) \end{aligned}$$

avînd drept bază inițială pe cea formată din vectorii e_1, e_2, P_5.

Tabelul 5

C_B	$c_j' \rightarrow$		0	0	0	0	0	$-\alpha$	$-\alpha$
$\downarrow$	Variabile bazice	P_0	P_1	P_2	P_3	P_4	P_5	e_1	e_2
$-\alpha$	x_6^a	2	2	1	1	-1	0	1	0
$-\alpha$	$\leftarrow x_7^a$	3	-1	$\boxed{2}$	-2	-2	0	0	1
0	x_5	9	3	-1	1	0	1	0	0
	$c_j' - z_j$	$z_0 = -5\alpha$	α	$+3\alpha$	$-\alpha$	-3α	0	0	0
$-\alpha$	$\leftarrow x_6^a$	1/2	$\boxed{5/2}$	0	2	0	0	1	$-1/2$
0	x_2	3/2	$-1/2$	1	-1	-1	0	0	1/2
0	x_5	21/2	5/2	0	0	-1	1	0	1/2
	$c - z_j$	$z_0 = -\alpha/2$	$5\alpha/2$	0	2α	0	0	0	$-3\alpha/2$
0	x_1	1/5	1	0	4/5	0	0	2/5	$-1/5$
0	x_2	8/5	0	1	$-3/5$	-1	0	1/5	2/5
0	x_5	10	0	0	-2	-1	1	-1	1
	$c_j' - z_j$	$z_0 = 0$	0	0	0	0	0	$-\alpha$	$-\alpha$

Scanned with OKEN Scanner

Faza I a luat sfîrşit. Deoarece $z_0 = 0$ şi nici o variabilă auxiliară nu este bazică, dispunem de o primă soluţie de bază a problemei iniţiale.

Pentru faza II primul tablou simplex (tabelul 6) se obţine din ultimul tablou de mai sus (tabelul 5) din care se elimină coloanele e_1 şi e_2 şi în care se introduc coeficienţii funcţiei obiectiv date.

Deci prima soluţie de bază are componentele

$$x_1 = 1/5, \quad x_2 = 8/5, \quad x_5 = 8/5, \quad x_3 = x_4 = 0.$$

Faza II

Tabelul 6

C_B	$c_j \rightarrow$		-3	7	$-1/2$	-5	-3
$\downarrow$	Variabile bazice	P_0	P_1	P_2	P_3	P_4	P_5
-3	$\leftarrow x_1$	$1/5$	1	0	$\boxed{4/5}$	0	0
7	x_2	$8/5$	0	1	$-3/5$	-1	0
-3	x_5	10	0	0	-2	-1	1
	$c_j - z_j$	$z_0 = -97/5$	0	0	$1/10$	-1	0
$-1/2$	x_3	$1/4$	$5/4$	0	1	0	0
7	x_2	$7/4$	$3/4$	1	0	-1	0
-3	x_5	$21/2$	$5/2$	0	0	-1	1
	$c_j - z_j$	$-155/8$	$-1/8$	0	0	-1	0

Deoarece toţi $c_j - z_j \leqslant 0$, ultima soluţie este soluţia optimă finită a problemei iniţiale

$$x_2^* = 7/4, \quad x_3^* = 1/4, \quad x_5^* = 21/2, \quad x_1^* = x_4^* = 0.$$

Scanned with OKEN Scanner

Exemplul 4. Fie programul

$$6x_1 - 2x_2 + x_3 - x_4 = 5,$$
$$2x_1 - \frac{1}{3}x_2 - x_3 + x_4 = 3,$$
$$3x_1 - x_2 + 2x_3 + 4x_4 = 2,$$
$$x_i \geqslant 0,\ i = 1, 2, 3, 4;$$
$$\max\ (2x_1 - x_2 + x_3 - x_4).$$

Matricea A nu conţine nici un vector unitar, deci este necesară introducerea a trei variabile auxiliare corespunzătoare vectorilor unitari e_1, e_2, e_3.

Faza I

Se rezolvă programul liniar

$$6x_1 - 2x_2 + x_3 - x_4 + x_5^a = 5,$$
$$2x_1 - \frac{1}{3}x_2 - x_3 + x_4 + x_6^a = 3,$$
$$3x_1 - x_2 + 2x_3 + 4x_4 + x_7^a = 2,$$
$$x_i \geqslant 0,\ i = 1, 2, 3, 4;\quad x_i^a \geqslant 0,\ i = 5, 6, 7;$$
$$\max\ (-\alpha x_5^a - \alpha x_6^a - \alpha x_7^a)$$

(tabelul 7)

Deoarece maximul pentru funcţia obiectiv în faza I este $-\alpha$ deci diferit de zero, tragem concluzia că programul dat nu are soluţii posibile.

Într-adevăr, acest lucru se poate verifica direct, rezolvînd sistemul de restricţii al problemei date.

Luînd ca determinant principal

$$\begin{vmatrix} 6 & -2 & 1 \\ 2 & -1/3 & -1 \\ 3 & -1 & 2 \end{vmatrix} = 3.$$

Soluţiile sistemului sînt

$$x_1 = 16/9 - \frac{14}{3}x_4,$$
$$x_2 = 8/3 - 16\ x_4,$$
$$x_3 = -1/3 - 3x_4,$$

Scanned with OKEN Scanner

Tabelul 7

C_B	$c_j' \rightarrow$		0	0	0	0	$-\alpha$	$-\alpha$	$-\alpha$
$\downarrow$	Variabile bazice	P_0	P_1	P_2	P_3	P_4	e_1	e_2	e_3
$-\alpha$	x_5^a	5	6	-2	1	-1	1	0	0
$-\alpha$	x_6^a	3	2	$-1/3$	-1	1	0	1	0
$-\alpha$	$\leftarrow x_7^a$	2	$\underline{\overline{\lvert 3 \rvert}}$	-1	2	4	0	0	1
	$c_j'-z_j$	$z_0 = -10\alpha$	11α	$-\frac{10}{3}\alpha$	2α	4α	0	0	0
$-\alpha$	x_5^a	1	0	0	-3	-9	1	0	-2
$-\alpha$	$\leftarrow x_6^a$	5/3	0	$\underline{\lvert 1/3 \rvert}$	$-7/3$	$-5/3$	0	1	$-2/3$
0	x_1	2/3	1	$-1/3$	2/3	4/3	0	0	1/3
	$c_j'-z_j$	$z_0 = -\frac{8}{3}\alpha$	0	$\alpha/3$	$-\frac{16}{3}\alpha$	$-\frac{32}{3}\alpha$	0	0	$-\frac{11}{3}\alpha$
$-\alpha$	x_5^a	1	0	0	-3	-9	1	0	-2
0	x_2	5	0	1	-7	-5	0	3	-2
0	x_1	7/3	1	0	$-5/3$	$-1/3$	0	1	$-1/3$
	$c_j'-z_j$	$z_0 = -\alpha$	0	0	-3α	-9α	0	$-\alpha$	-3α

Scanned with OKEN Scanner

Este uşor de văzut că x_3 şi x_4 nu pot fi simultan nenegative, deci programul nu are soluţii posibile.

Exemplul 5. Să se rezolve programul liniar

$$6x_1 - 2x_2 + x_3 - x_4 + 2x_5 = 4,$$
$$2x_1 - \frac{1}{3}x_2 - x_3 + x_4 = 3,$$
$$3x_1 - x_2 + 2x_3 + 4x_4 + x_5 = 2,$$
$$x_i \geqslant 0,\ i = 1, 2, \ldots, 5,$$
$$\max(-x_1 - 2x_2 - x_4 + 5x_5).$$

Pentru prima fază se introduc trei variabile artificiale.

Faza I

Se rezolvă programul

$$6x_1 - 2x_2 + x_3 - x_4 + 2x_5 + x^a = 4,$$
$$2x_1 - \frac{1}{3}x_2 - x_3 + x_4 + x_7^a = 3,$$
$$3x_1 - x_2 + 2x_3 + 4x_4 + x_5 + x_8^a = 2,$$
$$x_i \geqslant 0,\ i = 1, 2, \ldots, 5,\ x_k^a \geqslant 0,\ k = 6, 7, 8;$$
$$\max(-\alpha x_6^a - \alpha x_j^a - \alpha x_8^a).$$

(tabelul 8)

Prima fază ia sfîrşit aici, cu valoarea funcţiei obiectiv egală cu zero. Deoarece în bază figurează variabila artificială x_6^a, sîntem în cazul iii).

Înainte de a intra în faza II să facem remarca că în primul tablou, în alegerea elementului pivot, minimul rapoartelor $\frac{x_k^0}{z_{1k}}$, este atins pentru doi indici diferiţi,

$$\min\{4/6,\ 3/2,\ 2/3\} = 4/6 = 2/3.$$

Tabelul 8

C'_B	$c'_j \rightarrow$		0	0	0	0	0	$-\alpha$	$-\alpha$	$-\alpha$
$\downarrow$	Variabile bazice	P_0	P_1	P_2	P_3	P_4	P_5	e_1	e_2	e_3
$-\alpha$	x_6^a	4	6	-2	1	-1	2	1	0	0
$-\alpha$	x_7^a	3	2	$-1/3$	-1	1	0	0	1	0
$-\alpha$	$\leftarrow x_8^a$	2	$\underline{\overline{\lvert 3 \rvert}}$	-1	2	4	1	0	0	1
	$c'_j - z_j$	$z_0 = -9\alpha$	11α	$-\frac{10}{3}\alpha$	2α	4α	3α	0	0	0
$-\alpha$	x^a	0	0	0	-3	-9	0	1	0	-2
$-\alpha$	$\leftarrow x_2$	5/3	0	$\underline{\overline{\lvert 1/3 \rvert}}$	$-7/3$	$-5/3$	$-2/3$	0	1	$-2/3$
0	x_1	2/3	1	$-1/3$	2/3	4/3	1/3	0	0	1/3
	$c'_j - z_j$	$z_0 = -\frac{5}{3}\alpha$	0	$\alpha/3$	$-\frac{16}{3}\alpha$	$-\frac{32}{3}\alpha$	$-\frac{2}{3}\alpha$	0	0	$-\frac{1}{3}\alpha$
$-\alpha$	x_6^a	0	0	0	-3	-9	0	1	0	-2
0	x_2	5	0	1	-7	-5	-2	0	3	-2
0	x_1	7/3	1	0	$-5/3$	$-1/3$	$-1/3$	0	1	$-1/3$
	$c'_j - z_j$	0	0	0	-3	-9	0	0	$-\alpha$	-3α

Asemenea situaţii produc degenerarea soluţiei de bază, lucru observabil în tabloul al doilea.

Pentru faza II, se elimină vectorii P_3, P_4, e_2, e_3 deoarece le corespund, pe ultima linie, diferenţe negative.

Scanned with OKEN Scanner

Faza II

Tabelul 9

C_B	$c_j \rightarrow$		-1	-2	5	0
$\downarrow$	Variabile bazice	P_0	P_1	P_2	P_5	e_1
0	x_6^a	0	0	0	0	1
-2	x_2	5	0	1	-2	0
-1	x_1	$7/3$	1	0	$-1/3$	0
	$c_j - z_j$	$z_0 = -37/3$	0	0	$2/3$	0

Ne oprim aici deoarece $c_5 - z_5 = 2/3 > 0$, dar toți $z_{5k} \leqslant 0$. În concluzie, programul dat are soluție infinită.

5.4. Convergența algoritmului simplex. Ciclaje

Din expunerea principiilor generale care stau la baza metodei simplex, rezultă că fiecare etapă a algoritmului, produce o soluție de bază diferită de cea a etapei precedente.

Cum, pe de altă parte, numărul soluțiilor de bază este finit (cel mult C_m^n), s-ar putea trage concluzia, că procesul se încheie întotdeauna după un număr finit de iterații.

Se poate ivi însă o situație particulară, care pune sub semnul întrebării finitudinea algoritmului.

Geometric vorbind, mulțimea $\mathfrak{X}$ a soluțiilor posibile fiind un poliedru convex, cu vîrfurile, soluții de bază, s-ar putea ca iteratele algoritmului să ne conducă prin vîrfurile succesive ale unei fețe ale poliedrului, sau ale unui poliedru obținut prin secționarea poliedrului total, ajungînd la un moment dat din nou în vîrful inițial. Continuarea algoritmului ar deveni inutilă deoarece parcursul s-ar repeta la nesfîrșit, creindu-se așa numitul fenomen de ciclaj.

Scanned with OKEN Scanner

Ţinînd seama de faptul că şirul valorilor funcţiei obiectiv, asociate iteratelor succesive este nedescrescător, este uşor de remarcat că un asemenea fenomen nu se poate produce decît dacă valorile lui $f(X)$ sînt aceleaşi, în toate vîrfurile parcursului.

Or, din relaţia (11), rezultă că acest lucru se poate întîmpla doar dacă $\bar{x}_i = 0$ (i fiind indicele vectorului P_i care iese din bază în interaţia respectivă), adică doar în cazul soluţiilor degenerate.

Menţionăm că aceasta este o condiţie necesară pentru apariţia fenomenului de ciclaj, dar nu şi suficientă, în majoritatea cazurilor, soluţiile degenerate neducînd la ciclaje.

Pentru a se evita, totuşi, eventualitatea creerii ciclurilor în algoritmul simplex, s-au pus la punct metode preventive, care se aplică deci, în toate cazurile susceptibile de a produce ciclaj.

O asemenea metodă este metoda lexicografică.

Pe parcursul algoritmului, degenerarea apare întotdeauna cînd folosirea criteriului de alegere a vectorului ce iese din bază, minimul rapoartelor formate între componentele lui P_0 şi cele ale lui P_j(P_j intră în bază) este atins pentru doi sau mai mulţi indici diferiţi (vezi şi exemplul 5 tablourile 1 şi 2).

Fie atunci baza B, P_j vectorul, determinat, conform teoremei 2, ce trebuie introdus în bază, şi I submulţimea de indici pentru care:

$$\frac{x_i^0}{z_{ji}} = \min_{\substack{z_{jk}>0 \\ k\in\bar{R}}} \frac{x_k^0}{z_{jk}}, \text{ pentru orice } i \in I.$$

Fie $i_1 < i_2 < \ldots . < i_{n-m}$, indicii vectorilor nebazici.

Pentru determinarea vectorului care trebuie eliminat din bază, se alege acel indice al mulţimii I, pentru care:

$$\frac{z_{i_1 i}}{z_{ji}} = \min_{\substack{z_{jk}>0 \\ k\in T}} \frac{z_{i_1 k}}{z_{jk}}.$$

Dacă indicele i este unic, problema este rezolvată; în caz contrar, fie $I_1 \subset I$ submulţimea de indici, care realizează minimul precedent.

Scanned with OKEN Scanner

Se caută atunci $i \in I_1$, astfel ca

$$\frac{z_{i_2 i}}{z_{ji}} = \min_{\substack{z_{jk}>0 \\ k \in I_1}} \frac{z_{i_2 k}}{z_{jk}}.$$

Dacă nici de data aceasta indicele i nu este unic, fie $I_2 \subset I_1$ submulţimea indicilor care realizează ultimul indice. Se determină $i \in I_2$ astfel ca

$$\frac{z_{i_3 i}}{z_{ji}} = \min_{\substack{z_{jk}>0 \\ k \in I_2}} \frac{z_{i_3 k}}{z_{jk}}$$

şi aşa mai departe. Procesul se încheie, înainte de epuizarea tuturor indicilor $i_1, i_2, \ldots, i_{n-m}$, deoarece în caz contrar, proporţionalitatea a două sau mai multe linii din A ar implica rang $A < m$, contrar celor presupuse pînă în prezent.

Dăm în continuare un exemplu de program în care se poate produce fenomenul de ciclaj, dacă principiul expus aici nu se aplică.

Exemplul 6. Să se rezolve programul

$$x_1 + \frac{1}{4} x_4 - 8x_5 - x_6 + 9x_7 = 0,$$

$$x_2 + \frac{1}{2} x_4 - 12x_5 - \frac{1}{2} x_6 + 3x_7 = 0,$$

$$x_3 + x_6 = 1,$$

$$x_i \geqslant 0, \; i = 1, 2, \ldots, 7;$$

$$\max \left(\frac{3}{4} x_4 - 20x_5 + \frac{1}{2} x_6 - 6x_7 \right).$$

Întrucît matricea A conţine toţi vectorii unitari, dispunem de o primă soluţie de bază (tabelul 10).

Scanned with OKEN Scanner

Tabelul 10

C_B	$c_j \rightarrow$		0	0	0	3/4	−20	1/2	−6
↓	variabile bazice	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
0	$\leftarrow x_1$	0	1	0	0	$\underline{\boxed{1/4}}$	−8	−1	9
0	x_2	0	0	1	0	1/2	−12	−1/2	3
0	x_3	1	0	0	1	0	0	1	0
	$c_j - z_j$	$z_0 = 0$	0	0	0	3/4	−20	1/2	−6
3/4	x_4	0	4	0	0	1	−32	−4	36
0	$\leftarrow x_2$	0	−2	1	0	0	$\underline{\boxed{4}}$	3/2	−15
0	x_3	1	0	0	1	0	0	1	0
	$c_j - z_j$	$z_0 = 0$	−3	0	0	0	4	7/2	−33
3/4	$\leftarrow x_4$	0	−12	8	0	1	0	$\underline{\boxed{8}}$	−84
−20	x_5	0	−1/2	1/4	0	0	1	3/8	−15/4
0	x_3	1	0	0	1	0	0	1	0
	$c_j - z_j$	$z_0=0$	−1	−1	0	0	0	2	−18
1/2	x_6	0	−3/2	1	0	1/8	0	1	−21/2
−20	$\leftarrow x_5$	0	+1/16	−1/8	0	−3/64	1	0	$\underline{\boxed{3/16}}$
0	x_3	1	3/2	−1	1	−1/8	0	0	21/2
	$c_j - z_j$	$z_0 = 0$	+2	−3	0	−1/4	0	0	3

Scanned with OKEN Scanner

Tabelul 10 (continuare)

C_B	$c_j \rightarrow$		0	0	0	3/4	−20	1/2	−6
↓	variabile bazice	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
1/2	$\leftarrow x_6$	0	$\underline{\overline{\lvert 2 \rvert}}$	−6	0	−5/2	56	1	0
−6	x_7	0	1/3	−2/3	0	−1/4	16/3	0	1
0	x_3	1	−2	6	1	5/2	−56	0	0
	$c_j - z_j$	$z_0 = 0$	1	−1	0	1/2	−16	0	0
0	x_1	0	1	−3	0	−5/4	28	1/2	0
−6	$\leftarrow x_7$	0	0	$\underline{\overline{\lvert 1/3 \rvert}}$	0	1/6	−4	−1/6	1
0	x_3	1	0	0	1	0	0	1	0
	$c_j - z_j$	$z_0 = 0$	0	2	0	7/4	−44	−1/2	0
0	x_1	0	1	0	0	1/4	−8	−1	9
0	$\leftarrow x_2$	0	0	1	0	$\underline{\overline{\lvert 1/2 \rvert}}$	−12	−1/2	3
0	x_3	1	0	0	1	0	0	1	0
	$c_j - z_j$	$z_0 = 0$	0	0	0	3/4	−20	1/2	−6

Pe parcursul aplicării algoritmului nu am utilizat recomandările regulei lexicografice. Ultimul tablou este identic cu primul, corespunzînd aceleiași baze. Se observă că pe parcurs $f(X)$ a avut aceiași valoare (0).

Utilizarea regulii lexicografice, impune următoarele transformări.

Pornim din nou cu prima iterată.

Vectorul P_4 trebuie introdus în bază.

Pentru determinarea vectorului care o părăsește, observăm că $\min \frac{x_k^0}{z_{4k}} = \min \left\{ \frac{0}{1/4}, \frac{0}{1/2} \right\}$, este atins pentru indicii 1 și 2.

Scanned with OKEN Scanner

Primul vector nebazic care poate fi luat în considerare pentru a face discriminarea între cei doi indici, este P_7 (deoarece P_5 și P_6 au componentele din liniile 1 și 2 negative)

$$\min_{k=1,2} \frac{z_{7k}}{z_{4k}} = \min\left\{\frac{9}{1/4}, \frac{3}{1/2}\right\} = \frac{3}{1/2}, \text{ este atins pentru } k = 2.$$

Deci $z_{42} = 1/2$ se ia drept pivot (tabelul 11).

Tabelul 11

C_B	$c_j \rightarrow$		0	0	0	3/4	−20	1/2	−6
↓	variabile bazice	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
0	x_1	0	1	−1/2	0	0	−2	−3/4	15/2
3/4	x_4	0	0	2	0	1	−24	−1	6
0	$\leftarrow x_3$	1	0	0	1	0	0	[1]	0
	$c_j - z_j$	$z_0 = 0$	0	−3/2	0	0	−2	5/4	−21/2
0	x_1	3/4	1	−1/2	3/4	0	−2	0	15/2
3/4	x_4	1	0	2	1	1	−24	0	6
1/2	x_6	1	0	0	1	0	0	1	0
	$c_j - z_j$	$z_0 = 5/4$	0	−3/2	−5/4	0	−2	0	−21/2

Pornind de la același tabel inițial, am construit prima și a doua iterată simplex, utilizînd regula lexicografică. Ultimul tabel dă soluția optimă a programului.

$$x_1^* = 3/4,\ x_4^* = 1,\ x_6^* = 1,\ x_2^* = x_3^* = x_5^* = x_7^* = 0,$$

$$\max f(X) = 5/4.$$

Scanned with OKEN Scanner

În încheierea acestui paragraf, consacrat metodei simplex, amintim că în § 4 am făcut ipoteza rang $A = m$, ipoteză menţinută şi pe parcursul demonstraţiilor teoremelor fundamentale ale metodei simplex. Metoda celor două faze, face să cadă însă această restricţie, deoarece ea ne permite introducerea matricei unitate de ordin m, şi astfel matricea coeficienţilor în prima fază va avea sigur rangul m.

Rămîne deci, valabilă numai restricţia $m \leqslant n$.

În paragraful următor, vom vedea că abordarea problemei duale, duce la anularea şi acestei ultime ipoteze restrictive.

§ 6. DUALITATEA ÎN PROGRAMAREA LINIARĂ

6.1. Enunţarea problemei duale

Cu notaţiile din § 3, fie problema de programare liniară, pe care aici o vom numi *primală*,

$$
\begin{aligned}
&a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1,\\
&\cdots\cdots\cdots\cdots\cdots\\
&\cdots\cdots\cdots\cdots\cdots\\
&a_{l1}x_1 + a_{l2}x_2 + \ldots + a_{ln}x_n = b_l,\\
&a_{l+1}x_1 + a_{l+1,2}x_2 + \ldots + a_{l+1,n}x_n \leqslant b_{l+1},\\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
&a_{l+s,1}x_1 + a_{l+s,2}x_2 + \ldots + a_{l+s,n}x_n \leqslant b_{l+s},\\
&a_{l+s+1,1}x_1 + a_{l+s+1,2}x_2 + \ldots + a_{l+s+1,n}x_n \geqslant b_{l+s+1},\\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
&a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \geqslant b_m,
\end{aligned}
\tag{1}
$$

$l \geqslant 1,\ s \geqslant 1,\ l + s \leqslant m)$

$$
\begin{aligned}
&x_{i_1} \geqslant 0,\ x_{i_2} \geqslant 0,\ \ldots,\ x_{i_p} \geqslant 0,\\
&x_{i_{p+1}} \leqslant 0,\ x_{i_{p+2}} \leqslant 0,\ \ldots,\ x_{i_{p+r}} \leqslant 0,\quad p + r \leqslant n,\\
&x_{i_{p+r+1}},\ \ldots\ x_{i_n} \text{ oarecare};
\end{aligned}
\tag{2}
$$

$$
\max\ (\min)\ \{f(x_1, x_2, \ldots, x_n) = c_1x_1 + c_2x_2 + \ldots + c_nx_n\}. \tag{3}
$$

Scanned with OKEN Scanner

Se asociază acestei probleme un alt program liniar, numit *dual*, care constă în determinarea valorilor variabilelor reale $w_1, w_2, \ldots, w_m$ satisfăcînd condiţiile:

$$
\begin{aligned}
&-a_{1i_1}w_1 - a_{2i_1}w_2 - \ldots - a_{mi_1}w_m \geqslant -c_{i_1},\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&-a_{1i_p}w_1 - a_{2i_p}w_2 - \ldots - a_{mi_p}w_m \geqslant -c_{i_p},\\
&-a_{1i_{p+1}}w_1 - a_{2i_{p+1}}w_2 - \ldots - a_{mi_{p+1}}w_m \leqslant -c_{i_{p+1}},\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&-a_{1i_{p+r+1}}w_1 - a_{2i_{p+r}}w_2 - \ldots - a_{mi_{p+r}}w_m \leqslant -c_{i_{p+r}},\\
&-a_{1i_{p+r+1}}w_1 - a_{2i_{p+r+1}}w_2 - \ldots - a_{mi_{p+r+1}}w_m = -c_{i_{p+r+1}}\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&-a_{1i_n}w_1 - a_{2i_n}w_2 - \ldots - a_{mi_n}w_m = -c_{i_n};
\end{aligned}
\tag{4}
$$

$$
\begin{aligned}
&w_1, w_2, \ldots, w_l \text{ oarecare}\\
&w_{l+1} \leqslant 0,\ w_{l+2} \leqslant 0,\ \ldots,\ w_{l+s} \leqslant 0,\\
&w_{l+s+1} \geqslant 0,\ w_{l+s+2} \geqslant 0,\ \ldots,\ w_m \geqslant 0;
\end{aligned}
\tag{5}
$$

$$
\max(\min)\{g(w_1, w_2, \ldots, w_m) = -b_1w_1 - b_2w_2 - \ldots - b_mw_m\}. \tag{6}
$$

Regulile pentru formarea programului dual sînt deci:

— fiecărei variabile nenegative (nepozitive), a programului primal, îi corespunde în dual o inecuaţie $\geqslant$ ($\leqslant$);

— fiecărei variabile oarecare a programului primal îi corespunde în dual o ecuaţie;

— coeficienţii funcţiei obiectiv din problema primală sînt opuşii termenilor liberi din sistemul restricţiilor dualei;

— termenii liberi ai sistemului de restricţii primal sînt opuşii coeficienţilor funcţiei obiectiv duale;

— fiecărei restricţii de forma $\geqslant$, $\leqslant$, $=$, din programul primal, îi corespunde în dual o variabilă nenegativă, nepozitivă, respectiv, oarecare;

— matricea coeficienţilor din sistemul de restricţii dual este opusa transpusei matricei coeficienţilor din primal.

Scanned with OKEN Scanner

Se poate uşor remarca că programul dual corespunzător programului (4), (5), (6) este chiar (1), (2), (3), deci, duala dualei este problema primală.

În particular dacă programul primal este dat sub forma canonică, atunci şi cel dual va apare tot sub o formă asemănătoare.

Folosind notaţiile vectoriale, şi schimbînd semnul coeficienţilor din dual, cînd este cazul, obţinem:

Programul primal.

(1^c) $\quad AX \leqslant b,$

(2^c) $\quad X \geqslant 0$

(3^c) $\quad \max CX.$

Programul dual.

(4^c) $\quad WA \geqslant C,$

(5^c) $\quad W \geqslant 0$

(6^c) $\quad \min Wb';$

aici, W este vectorul linie $(w_1, w_2, \ldots, w_m)$.

Dacă programul primal este adus la forma standard

(1^s) $\quad AX = b,$

(2^s) $\quad X \geqslant 0,$

(3^s) $\quad \max CX.$

programul dual va fi

(4^s) $\quad WA \geqslant C,$

(5^s) $\quad W$ oarecare,

(6^s) $\quad \min Wb.$

Forma programului dual nu este însă cea standard, lipsind ipoteza de nenegativitate pentru W.

Scanned with OKEN Scanner

Intre cuplurile de probleme duale nu există numai o legătură formală ci și o strînsă interdependență a soluțiilor lor. Teoremele fundamentale ale dualității vor preciza aceasta.

Să considerăm cuplul de programe duale (1^c), (2^c), (3^c), și (4^c), (5^c), (6^c).

Lemă. *Dacă X, W, constituie o pereche de soluții posibile pentru programul primal, respectiv, dual, atunci are loc inegalitatea*

$$CX \leqslant Wb. \tag{7}$$

Într-adevăr, W ca soluție a programului dual satisface sistemul de restricții (4^c), deci

$$C \leqslant WA$$

(inegalitatea vectorială, de aici trebuie interpretată în sensul inegalității între toate componentele)

Înmulțind la dreapta cu X, soluție posibilă, cu toate componentele nenegative, obținem

$$CX \leqslant WAX.$$

Dar, X satisface (1^c) și astfel

$$CX \leqslant Wb.$$

Ca o consecință imediată a acestei leme se poate demonstra

Teorema 1. Dacă pentru un cuplu X^*, W^* *de soluții posibile ale celor două probleme duale are loc egalitatea*

$$CX^* = W^*b \tag{8}$$

atunci X^* este soluție optimă a programului primal, iar W^* este soluție optimă a programului dual.

Pentru demonstrație, presupunem, prin absurd, că X^* nu este optimul programului (1^c), (2^c) (3^c). Atunci există X' soluție posibilă astfel ca

$$CX' > CX^*$$

Scanned with OKEN Scanner

și de aici, în virtutea egalității (8),

$$CX' > Wb,$$

ceea ce ar contrazice rezultatul lemei.

Cea de a doua afirmație a teoremei se demonstrează analog.

Din lemă se poate ușor deduce că dacă programul primal are optim infinit, atunci programul dual nu are soluții posibile.

Într-adevăr, o consecință a inegalității (7) este

$$\max CX \leqslant \min Wb.$$

Or, dacă prima problemă are optimul infinit, ar rezulta de aici că și $\min Wb = +\infty$, ceea ce pentru o problemă de minim este imposibil.

Într-un mod cu totul asemănător se poate arăta că în eventualitatea în care programul dual are optim infinit ($-\infty$), atunci programul inițial nu admite soluții posibile.

Aceste rezultate pot fi completate, ajungîndu-se la formularea teoremelor fundamentale ale dualității.

Teoremă (de existență). *Pentru un cuplu de programe liniare duale, sînt posibile doar următoarele alternative:*

i) *niciunul din cele două programe nu admite soluții posibile;*

ii) *un program are optim infinit, iar celălalt nu admite soluții posibile;*

iii) *ambele programe admit soluții optime finite.*

Teoremă (de optim în dualitate). *Pentru un cuplu de programe liniare duale, condiția necesară și suficientă ca soluția posibilă X^* (W^*) a programului primal (dual), să fie soluție optimă este să existe o soluție posibilă a programului dual (primal) $W(X)$, astfel ca*

$$CX^* = Wb, \quad (CX = W^*b).$$

Se poate deduce ușor și alt rezultat care pune în evidență o proprietate interesantă, cu importante consecințe practice, a perechilor de soluții optime duale.

Scanned with OKEN Scanner

Teorema 2. *Pentru un cuplu de programe liniare duale condiția necesară și suficientă ca soluțiile posibile X^*, W^* să constituie o pereche de soluții optime este:*

$$W^*(b - AX^*) = 0, \tag{9}$$
$$(W^*A - C)X^* = 0.$$

Într-adevăr, sumînd cele două relații membru cu membru avem

$$W^*b = CX^* = 0$$

ceea ce conform teoremei 1, implică optimalitatea soluțiilor.

Reciproc, să presupunem că X^*, W^* sînt soluții optime ale celor două probleme.

Atunci, din (1^c), (2^c), (4^c), (5^c) rezultă

$$W^*(b - AX^*) \geqslant 0, \tag{10}$$
$$(W^*A - C)X^* \geqslant 0.$$

Pe de altă parte

$W^*(b - AX^*) + (W^*A - C)X^* = W^*b - W^*AX^* + W^*AX^* - CX^* = W^*b - CX^* = 0$ în virtutea teoremei de optim.

Or, suma a două numere nenegative nu poate fi nulă decît dacă ambii termeni sînt nuli. De aici, (10) se transformă în (9).

Interpretarea egalităților (9) este următoarea:

— dacă soluția optimă X^* a problemei primale satisface cu inegalitate strictă, restricția i din sistemul (1^c), ($i = 1, 2, \ldots, m$), atunci componenta w_i^* a soluției programului dual este nulă;

— dacă soluția optimă W^* a problemei duale satisface cu inegalitate strictă restricția $j (j = 1, 2, \ldots, n)$, a sistemului (4^c), atunci componenta x_j^* a soluției programului primal este nulă;

— dacă componenta w_i^*, ($i = 1, 2, \ldots, m$) a soluției optime a programului dual este strict pozitivă, atunci soluția X^* a programului primal satisface cu egalitate restricția i a sistemului (1^c);

Scanned with OKEN Scanner

— dacă componenta x_j^* $(j = 1, 2, \ldots, n)$ a soluției optime a programului primal este strict pozitivă, atunci soluția optimă W^* a programului dual satisface cu egalitate restricția j a sistemului (4c).

6.3. Aflarea soluției programului dual din tabloul simplex corespunzător soluției optime a programului primal

Să presupunem că am rezolvat programul primal (1c), (2c), (3c). Evident pentru aceasta a fost necesară introducerea variabilelor de compensare nenegative y_k $(k = 1, 2, \ldots, m)$, pentru transformarea sistemului de inecuații (2c) în sistemul de ecuații.

(11) $$AX + EY = b$$

cu $$Y = \begin{Vmatrix} y_1 \\ \vdots \\ y_m \end{Vmatrix} \geqslant 0.$$

Dacă presupunem $b \geqslant 0$, atunci vectorii Q_k, corespunzători variabilelor de compensare, sînt chiar vectorii unitari e_k.

Fie X^* o soluție optimă a programului.

Aceasta înseamnă că

(12) $$c_j - z_j \leqslant 0, \qquad j = 1, 2, \ldots, n + m$$

unde, cu notațiile din § 5,

$$z_j = C_B Z_j, \qquad j = 1, 2, \ldots, n.$$

De asemenea corespunzător variabilelor noi introduse vom avea

$$c_j = 0, \qquad j = n + 1, \ldots, n + m,$$

$$z_j = C_B Q_j, \qquad j = n + 1, \ldots, n + m.$$

Trebuie să facem observația că în baza optimală B pot apare și vectorii Q_j.

Cum, Z_j are aici semnificația reprezentării vectorului P_j (sau Q_j), în baza B, rezultă clar că

$$Z_j = B^{-1}P_j \qquad (j = 1, 2, \ldots, n)$$

$$Z_j = B^{-1}Q_j, \qquad j = n + 1, \ldots, n + m.$$

Scanned with OKEN Scanner

Atunci (12) se transformă în

$$C_B B^{-1} P_j \geqslant c_j, \quad j = 1, 2, \ldots, n,$$

(13)

$$C_B B^{-1} e_j \geqslant 0, \quad j = n+1, n+2, \ldots, n+m.$$

Relaţiile (13) nu spun altceva decît că vectorul $C_B B^{-1}$ este soluţie posibilă a programului dual (4^c), (5^c), (6^c).

Într-adevăr, relaţia (4^c) se poate scrie sub forma

$$W \,||P_1, P_2, \ldots, P_n|| \geqslant C$$

sau

$$WP_j \geqslant c_j, \qquad j = 1, 2, \ldots, n.$$

Or, conform primei relaţii (13) vectorul linie m- dimensional $C_B B^{-1}$ are această proprietate, pe cînd cea de a doua relaţie (13) indică nenegativitatea tuturor componentelor sale.

De asemenea putem scrie

(14) $$g(C_B B^{-1}) = C_B B^{-1} b = C_B P_0 = f(X^*).$$

Atunci, din (13), (14) şi teorema 1, putem trage concluzia că $C_B \cdot B^{-1}$ este soluţie optimă a programului dual.

Se pot face unele observaţii, în baza teoremei 2.

Am notat aici cu X^*, soluţia optimă a programului originar (1^c), (2^c). (3^c). În realitate metoda simplex se aplică programului standard în care (1^c) se înlocuieşte cu (11). Soluţia acestui din urmă program este $\left\| \begin{matrix} X^* \\ Y^* \end{matrix} \right\|$, conţinînd $n + m$ componente.

Se poate întîmpla deci, ca printre variabilele bazice ale tabloului simplex optimal să apară şi variabile de compensare.

Este clar că $y_k^* > 0$, înseamnă că X^* satisface cu inegalitate strictă restricţia k din (1^c), şi atunci, ţinînd seama de interpretarea teoremei 3, $w_k^* = 0$.

În continuare vom aplica, pe un exemplu numeric, rezultatelele deduse în prezenta secţiune.

Scanned with OKEN Scanner

Exemplul 1. Fie programul liniar

$$\begin{aligned} 2x_1 + x_2 + x_3 + x_4 &\leqslant 2, \\ -x_1 + 2x_2 - 2x_3 + 2x_4 &\leqslant 3, \\ 3x_1 - x_2 + x_3 &\leqslant 9, \\ x_i \geqslant 0,\ i &= 1, 2, 3, 4; \end{aligned}$$

$$\max\left(-3x_1 + 7x_2 - \frac{1}{2}x_3 - 5x_4\right).$$

Aducerea la forma standard se face sumînd în membrul stîng al fiecărei inegalităţi cîte o variabilă de compensare nenegativă;

$$\begin{aligned} 2x_1 + x_2 + x_3 + x_4 + y_1 &= 2, \\ -x_1 + 2x_2 - 2x_3 + 2x_4 + y_2 &= 3, \\ 3x_1 - x_2 + x_3 + y_3 &= 9, \\ y_i \geqslant 0, \quad i &= 1, 2, 3 \end{aligned}$$

Soluţia optimă finită a primalei este deci:

$$x_3^* = 1/4,\ x_2^* = 7/4,\ x_1^* = x_4^* = 0;$$

$$f(X^*) = 97/8.$$

(tabelul 12)

Din teorema fundamentală a dualităţii rezultă deci că şi programul dual va avea soluţie optimă finită, valoarea funcţiei obiectiv fiind tot 97/8.

C_BB^{-1} reprezintă produsul dintre vectorul format de coeficienţii bazici şi inversa matricei de bază din ultimul tablou simplex, care apare aici pe coloanele corespunzătoare vectorilor unitari din primul tablou.

Deci, C_BB^{-1} se citeşte pe ultima linie luînd cu semn schimbat valorile de pe coloanele Q_1, Q_2, Q_3.

Soluţia optimă a dualei este deci

$$w_1^* = 13/4, \qquad w_2^* = 15/8, \qquad w_3^* = 0.$$

(ultima componentă rezultă nulă şi din observaţia $y_3 = 21/2 > 0$).

Scanned with OKEN Scanner

Tabelul 12

C_B	$c_j \rightarrow$		-3	7	$-1/2$	-5	0	0	0
$\downarrow$	variabile bazice	P_0	P_1	P_2	P_3	P_4	Q_1	Q_2	Q_3
0	y_1	2	2	1	1	1	1	0	0
0	$\leftarrow y_2$	3	-1	$\boxed{2}$	-2	2	0	1	0
0	y_3	9	3	-1	1	0	0	0	1
	$c_j - z_j$	$z_0 = 0$	-3	7	$-1/2$	-5	0	0	0
0	$\leftarrow y_1$	1/2	5/2	0	$\boxed{2}$	0	1	$-1/2$	0
7	x_2	3/2	$-1/2$	1	-1	1	0	1/2	0
0	y_3	21/2	5/2	0	0	1	0	1/2	1
	$c_j - z_j$	$z_0 = 21/2$	1/2	0	13/2	-12	0	$-7/2$	0
$-1/2$	$\leftarrow x_3$	1/4	5/4	0	1	0	1/2	$-1/4$	0
7	x_2	7/4	3/4	1	0	1	1/2	1/4	0
0	y_3	21/2	5/2	0	0	1	0	1/2	1
	$c_j - z_j$	$z_0 = 97/8$	$-61/8$	0	0	-12	$-13/4$	$-15/8$	0

Cum duala are forma

$$
\begin{aligned}
2w_1 - w_2 + 3w_2 &\geqslant -3, \\
w_1 + 2w_2 - w_3 &\geqslant 7, \\
w_1 - 2w_2 + w_3 &\geqslant -1/2, \\
w_1 + 2w_2 &\geqslant -5, \\
w_i \geqslant 0, \quad i &= 1, 2, 3; \\
\min \{g(W) = 2w_1 + 3w_2 + 9w_3\}&.
\end{aligned}
$$

Scanned with OKEN Scanner

Se poate uşor verifica că W^*, determinat mai sus, este soluţia posibilă a dualei. Pentru optimalitatea sa este suficientă verificarea egalităţii

$$g(W^*) = f(X^*) = 97/8.$$

6.4. Rezolvarea unui program liniar cu ajutorul dualului său

Din cele expuse anterior, se desprinde ideia că deducerea soluţiei unui program din cea a dualului său se face fără mari dificultăţi.

De aceea este recomandabil ca ori de cîte ori, rezolvarea programului dual se anunţă mai facilă decît cea a programului dat, să se rezolve dualul şi să se deducă apoi de aici soluţia căutată.

Am arătat la sfîrşitul § 5, că singura ipoteză restrictivă în care putem aplica algoritmul simplex, pe baza rezultatelor expuse, era $m \leqslant n$.

În caz contrar, adică $m > n$, putem recurge la programul dual.

Într-adevăr dacă matricea A are dimensiunile $(m \times n)$ atunci sistemul de restricţii din dual va avea n relaţii. Astfel forma dualului se pretează a fi încadrată în modelul general expus în paragrafele precedente.

Vom ilustra acest procedeu printr-un exemplu simplu.

Exemplul 2. Să se rezolve programul

$$\begin{aligned} 4x_1 + 3x_2 &\geqslant 24, \\ 5x_1 + 9x_2 &\geqslant 45, \\ 3x_1 - 4x_2 &\geqslant -44, \\ -x_1 &\geqslant -8, \\ x_1,\ x_2 &\geqslant 0; \\ \min\ (x_1 + x_2)&. \end{aligned}$$

Dualul are forma

$$\begin{aligned} 4w_1 + 5w_2 + 3w_3 - w_4 &\leqslant 1, \\ 3w_1 + 9w_2 - 4w_3 &\leqslant 1, \\ w_i \geqslant 0,\quad i = 1,\ 2,\ 3,\ 4&; \\ \max\ (24w_1 + 45w_2 - 44w_3 - 8w_4)&. \end{aligned}$$

Scanned with OKEN Scanner

Se aduce la forma standard

$$4w_1 + 5w_2 + 3w_3 - w_4 + y_1 = 1,$$
$$3w_1 + 9w_2 - 4w_3 + y_2 = 1,$$

cu $y_1 \geqslant 0, \quad y_2 \geqslant 0.$

Se aplică metoda simplex (tabelul 13).

Tabelul 13

C_B	c_j		24	45	−44	−8	0	0
↓	variabile bazice	P_0	P_1	P_2	P_3	P_4	Q_1	Q_2
0	y_1	1	4	5	3	−1	1	0
0	← y_2	1	3	|9|	−4	0	0	1
	$c_j - z_j$	$z_0 = 0$	24	45	−44	−8	0	0
0	← y_1	4/9	|7/3|	0	47/9	−1	1	− 5/9
45	w_2	1/9	1/3	1	− 4/9	0	0	1/9
	$c_j - z_j$	$z_0 = 5$	9	0	−24	−8	0	−5
24	w_1	4/21	1	0	47/21	−3/7	3/7	− 5/21
45	w_2	1/21	0	1	−25/21	1/7	−1/7	4/21
	$c_j - z_j$	z_0=45/7	0	0	−309/7	− 29/7	−27/7	−20/7

S-a obţinut soluţia optimă a dualei.

De aici se poate deduce soluţia optimă a problemei iniţiale.

Pe ultima linie pe coloanele Q_1, Q_2, sînt componentele, cu semn schimbat, al soluţiei căutate.

$$x_1^* = 27/7, \quad x_2^* = 20/7.$$

Problema iniţială a fost rezolvată direct, geometric în § 2., exemplul 2.

Scanned with OKEN Scanner

§. 7 ALGORITMUL SIMPLEX DUAL

7.1. Preliminarii

Exploatarea proprietăţilor fundamentale ale dualităţii, a sugerat mai multe metode de rezolvare a perechii de programare duale, sau a unuia singur dintre ele, plecînd de la acea problemă a cărei expresie algebrică este cea mai simplă. La sfîrşitul paragrafului precedent am expus un asemenea procedeu în care rezolvarea problemei primale se poate face prin aplicarea algoritmului simplex programului dual.

Algoritmul simplex dual este o metodă de rezolvare a programului primal în care algoritmul simplex suportă anumite modificări care speculează legătura intimă dintre soluţiile celor două probleme duale.

Fără a intra în amănunte, vom face o expunere succintă a acestui algoritm, limitîndu-ne strict la elementele necesare în capitolul IV.

Am văzut în § 6 (relaţiile (12), (13)), că, dacă pentru un X, diferenţele $c_j - z_j$ sînt nepozitive, programul dual admite soluţie posibilă vectorul $C_B B^{-1}$.

Această observaţie joacă un rol important în elaborarea algoritmului simplex dual.

Acesta constă, în esenţă, dintr-un şir de iterate pentru problema primală, care operează nu cu soluţii de bază ci cu aşa numite soluţii *dual-posibile*.

Definiţie. *X va fi numit soluţie dual-posibilă a programului liniar*

$$AX = b, \tag{1}$$

$$X \geqslant 0; \tag{2}$$

$$\max \{f(X) = CX\} \tag{3}$$

dacă, vectorii P_k corespunzători componentelor nenule sînt liniar-independenţi şi

$$c_j - z_j \leqslant 0, \quad \text{pentru } j = 1, 2, \ldots, n. \tag{4}$$

Observăm că potrivit acestei definiții, X nu este obligatoriu soluție de bază a programului primal, nefiind supus restricției de nenegativitate.

Libertatea de a opera cu vectorii X nesupuși condiției de nenegativitate, permite îndepărtarea restricției $b \geqslant 0$, și facilitează de multe ori inițierea algoritmului simplex, reducînd numărul variabilelor artificiale.

Datorită lui (4), devine evident faptul că o soluție dual-posibilă devine soluție optimă a programului primal, dacă are toate componentele nenegative.

7.2. Descrierea algoritmului

Ultima observație este utilizată de algoritmul simplex dual, care propune un procedeu de construcție a iterațiilor succesive, de asemenea manieră, încît numărul componentelor negative ale soluțiilor dual posibile corespunzătoare să descrească treptat pînă la dispariția completă.

Să presupunem că vectorii $P_j (j \in \overline{R})$ formează o bază B în spațiul restricțiilor și fie $\overline{X} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ astfel că

$$\sum_{j \in \overline{R}} \bar{x}_j P_j = P_0, \tag{5}$$

$$\bar{x}_j = 0, \quad \text{pentru } j \in R.$$

Algoritmul pleacă de la tabloul simplex corespunzător bazei B, presupunînd că $\overline{X}$ este soluție dual-posibilă.

Se testează optimalitatea lui $\overline{X}$.

Dacă

$$P_0 = B^{-1} b \geqslant 0, \tag{6}$$

atunci $\overline{X}$ este soluție optimă a programului (1) — (3).

Altfel, se pot ivi două posibilități:

i) dacă există un indice $i \in \overline{R}$ cu $\bar{x}_i < 0$, pentru care $z_{ki} \geqslant 0$, $(k = 1, 2, \ldots, n)$, atunci programul dual are optim infinit și deci (teorema fundamentală a dualității) cel primal nu are soluții posibile.

ii) În caz contrar se determină un vector $P_i (i \in \overline{R})$ care va părăsi baza B, și un vector $P_j (j \in R)$ care va fi introdus în

Scanned with OKEN Scanner

bază, înlocuirea avînd ca efect eliminarea unei componente negative a soluției dual-posibile $\overline{X}$.

i și j se determină astfel:

Se caută

(7) $$\tilde{x}_i = \min \tilde{x}_k, \quad \tilde{x}_k < 0$$

Cu i determinat ca mai sus se caută

(8) $$\frac{c_j - z_j}{z_{ij}} = \min_{z_{ik}<0} \frac{c_k - z_k}{z_{ik}}.$$

Cu aceste elemente, prima iterată a algoritmului constă în transformarea tuturor elementelor vechiului tablou simplex după regula dreptunghiului cu z_{ij} pivot.

Pentru noua soluție dual-posibilă $\overline{X}'$, corespunzătoare bazei B' obținută prin înlocuirea lui P_i cu P_j, se aplică testul de optimalitate (6).

În cazul cînd răspunsul este afirmativ, ea va constitui soluția optimă căutată. Dacă răspunsul este negativ și alternativa i) nu s-a realizat, se inițiază o nouă iterată, ținînd cont de (7) și (8).

Exemplul 1. Să se rezolve programul

$$x_1 - x_3 + x_4 - \frac{1}{2} x_5 = -2$$

$$x_2 - x_3 - x_4 + x_5 = 1,$$

$$x_i \geqslant 0. \quad i = 1, 2, \ldots, 5;$$

$$\max (-x_3 - x_4 - x_5).$$

Matricea A a coeficienților din sistemul liniar de ecuații este

$$A = \left\| \begin{matrix} 1 & 0 & -1 & 1 & -1/2 \\ 0 & 1 & -1 & -1 & 1 \end{matrix} \right\|.$$

Se observă prezența celor doi vectori unitari, $P_1 = e_1$, $P_2 = e_2$.

Deoarece $b_1 = -2 < 0$, aplicarea algoritmului simplex nu este posibilă fără anumite transformări, care vor avea ca efect

Scanned with OKEN Scanner

și necesitatea introducerii unei variabile artificiale. Vom aplica, în consecință algoritmul simplex dual. Primul tabel simplex corespunzător bazei $B = (P_1, P_2)$ este tabelul 14).

Tabelul 14

C_B	$c_j \rightarrow$		0	0	−1	−1	−1
↓	variabile bazice	P_0	P_1	P_2	P_3	P_4	P_5
0	$\leftarrow x_1$	−2	1	0	$\boxed{-1}$	1	−1/2
0	x_2	1	0	1	−1	−1	1
	$c_j - z_j$	$z_0 = 0$	0	0	−1	−1	−1

Deoarece $c_j - z_j \leqslant 0$, $X = (-2, 1, 0, 0, 0)$ este o soluție dual posibilă.

Evident $i = 1$ $(x_1 = -2 < 0)$, Pentru j, alegem

$$\min\left\{\frac{-1}{-1}, \frac{-1}{-1/2}\right\} = \frac{-1}{-1}.$$

Deci intră în bază P_3 în locul lui P_1; $z_{31} = -1$ este pivot. Prima iterată are ca efect completarea tabelului 15

Tabelul 15

C_B	$c_j \rightarrow$		0	0	−1	−1	−1
↓	Variabile bazice	P_0	P_1	P_2	P_3	P_4	P_5
−1	x_3	2	−1	0	1	−1	1/2
0	x_2	3	−1	1	0	−2	3/2
	$c_j - z_j$	$z_0 = -2$	−1	0	0	−2	−1/2

Scanned with OKEN Scanner

Cum noile componente ale soluției dual posibole sînt nenegative, algoritmul se oprește, odată cu atingerea optimului finit. Deci

$$x_2^* = 3, \qquad x_3^* = 2, \qquad x_1^* = x_4^* = x_5^* = 0;$$
$$f(X^*) = -2.$$

Observație. Spre deosebire de situația întîlnită pe parcursul iterațiilor simplex, de această dată, valorile lui $f(X)$, *descresc*, pînă la atingerea optimului căutat.

Exemplul 2. Fie programul

$$\begin{aligned} &6x_1 - x_2 \qquad\quad - x_4 = +5,\\ &2x_1 - \frac{1}{3}x_2 - x_3 \qquad = 3,\\ &3x_1 \qquad\quad + x_5 = 2,\\ &x_i \geqslant 0, \quad i = 1, 2, \ldots, 5;\\ &\max\ (2x_1 - x_2 + x_3 - x_4). \end{aligned}$$

Înmulțind cu -1, primele două ecuații, obținem vectorii unitari în matricea coeficienților, cu prețul negativizării termenilor liberi.

Sistemul de restricții va fi deci:

$$\begin{aligned} &-6x_1 + x_2 \qquad\quad + x_4 = -5,\\ &-2x_1 + \frac{1}{3}x_2 + x_3 \qquad = -3,\\ &3x_1 \qquad\quad + x_5 = \quad 2. \end{aligned}$$

Primul tabel simplex (16) va fi scris corespunzător bazei $B = (P_4, P_3, P_5)$.

$c_j - z_j \leqslant 0$, pentru $j = 1, 2, \ldots, 5$, deci $X = (0, 0, -3, -5, 2)$ este soluție dual-posibilă.

Se inițiază prima iterată. Determinăm vectorul ce iese din bază, corespunzător componentei $x_4 = -5$.

P_j corespunde raportului $\frac{-2}{-6}$ (unic).

Scanned with OKEN Scanner

Tabelul 16

C_B	$c_j \rightarrow$		2	−1	1	−1	0
↓	variabile bazice	P_0	P_1	P_2	P_3	P_4	P_5
−1	x_4	−5	−6	1	0	1	0
1	x_3	−3	−2	1/3	1	0	0
0	x_5	2	3	0	0	0	1
	$c_j - z_j$	$z_0 = 2$	−2	−1/3	0	0	0

Intră deci P_1, iese P_4; element pivot $z_{14} = -6$.

În următoarele două tabele (17) dăm iteratele 1 și 2 ale algoritmului.

Tabelul 17

C_B	$c_j \rightarrow$		2	−1	1	−1	0
↓	Variabile bazice	P_0	P_1	P_2	P_3	P_4	P_5
2	x_1	5/6	1	−1/6	0	−1/6	0
1	$\leftarrow x_3$	−4/3	0	0	1	$\boxed{-1/3}$	0
0	x_5	−1/2	0	1/2	0	1/2	1
	$c_j - z_j$	$z_0 = 1/3$	0	−2/3	0	−1/3	0
2	x_1	3/2	1	−1/6	−1/2	0	0
−1	x_4	4	0	0	−3	1	0
0	x_5	−5/2	0	1/2	3/2	0	1
	$c_j - z_j$	$z_0 = -1$	0	−2/3	−1	0	0

Scanned with OKEN Scanner

Deoarece corespunzător componentei negative $x_5 = -5/2$, toți $z_{j5} \geqslant 0$, tragem concluzia că programul dat nu are soluții posibile.

§ 8. PROBLEMA TRANSPORTURILOR

În § 1, exemplele V și VI au pus în evidență o formă particulară de problemă de programare liniară pe care am numit-o problema transporturilor.

În practică un asemenea gen de problemă se întîlnește atunci cînd un anumit produs aflat în cantitățile $a_1, a_2, \ldots, a_m$ în m puncte diferite $A_1, A_2, \ldots, A_m$ numite surse, trebuie transportat în alte n puncte $B_1, B_2, \ldots, B_n$ numite destinații, unde este cerut în cantitățile $b_1, b_2, \ldots, b_n$ urmărindu-se totodată minimizarea cheltuielilor totale de transport (date fiind prețurile unitare de transport c_{ij} de la sursa i la destinația j).

Deși probleme de aceeași formă matematică se pot pune și plecînd de la procese economice de altă natură (vezi exemplul VI, § 1), vom folosi întotdeauna terminologia sugerată de modelul expus anterior.

Primele rezultate privind problema transporturilor au fost obținute de Hitchcock, Kantorovici, și Koopmans. Ulterior, cercetările în acest domeniu au fost diversificate, obținîndu-se metode eficiente de rezolvare. Dantzig a arătat că o asemenea problemă poate fi integrată în modelul general al problemelor programării liniare, și ca atare metoda simplex poate fi adaptată condițiilor particulare, dînd drept rezultat, un procedeu de rezolvare principial identic cu cel folosit în cazul general.

Formularea matematică a problemei transporturilor este următoarea:

În condițiile

$$(1) \qquad \sum_{j=1}^{n} x_{ij} \leqslant a_i, \qquad i = 1, 2, \ldots, m;$$

$$(2) \qquad \sum_{i=1}^{m} x_{ij} \geqslant b_j, \qquad j = 1, 2, \ldots, n;$$

$$(3) \qquad x_{ij} \geqslant 0, \quad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n;$$

Scanned with OKEN Scanner

(4) să se minimizeze

$$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

unde parametrii problemei satisfac condiţiile

(5) $$a_i \geqslant 0,\ b_j \geqslant 0,\ c_{ij} \geqslant 0, \sum_{i=1}^{m} a_i \geqslant \sum_{j=1}^{n} b_j.$$

Într-adevăr dacă x_{ij} va reprezenta cantitatea de produs ce trebuie transportată de la sursa A_i la destinaţia B_j, relaţiile (1) impun condiţia ca totalul transportat de la fiecare sursă să nu depăşească existentul, relaţiile (2) impun satisfacerea cererii, (4) cere minimizarea cheltuielilor totale de transport.

Condiţiile (3) şi (5) apar naturale în contextul concret al problemei.

Prin transformări elementare orice problemă de tipul (1)— (5) poate fi adusă la forma:

(1′) $$\sum_{j=1}^{n} x_{ij} = a_i, \qquad i = 1, 2, \ldots, m,$$

(2′) $$\sum_{i=1}^{m} x_{ij} = b_j, \qquad j = 1, 2, \ldots, n,$$

(3) $$x_{ij} \geqslant 0, \qquad i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n;$$

(4) $$\min \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}\, x_{ij}$$

iar asupra parametrilor se poate face supoziţia

(5′) $$a_i \geqslant 0, \quad b_j \geqslant 0,\ c_{ij} \geqslant 0, \qquad \sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j.$$

Egalitatea din (5′) poate fi realizată introducînd o destinaţie fictivă căreia să-i fie adresat surplusul de produs existent pe ansamblul surselor.

În mod obişnuit datele problemei se prezintă sub forma unui tabel (18).

Scanned with OKEN Scanner

Tabelul 18

Destinația / Sursa	1	2	...	j	...	n	
1	c_{11}	c_{12}	...	c_{1j}	...	c_{1n}	a_1
2	c_{21}	c_{22}	...	c_{2j}	...	c_{2n}	a_2
...	. . . . .						...
i	c_{i1}	c_{i2}	...	c_{ij}	...	c_{in}	a_i
...	. . . . .						...
m	c_{m1}	c_{m2}	...	c_{mj}	...	c_{mn}	a_m
	b_1	b_2	...	b_j	....	b_n	

Să notăm prescurtat cu (T) problema definită de grupul de restricții (1'), (2'), (3), (4), în condițiile (5').

Putem formula următorul rezultat:

Teorema 1. *Problema* (T) *admite cel puțin o soluție posibilă.*

Într-adevăr, notînd $\sum_{j=1}^{m} a_i = \sum_{j=1}^{n} b_j = S$, este ușor de verificat că $x_{ij} = \frac{a_i b_j}{S}$, $(i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, n)$, verifică relațiile (1'), (2') și (3), constituind deci o soluție posibilă.

Cum rezultatele generale stabilite în § 4 sînt aplicabile și programului particular (T), va rezulta din teorema precedentă și existența soluțiilor de bază.

Mai mult, deoarece un program liniar oarecare nu poate să se situeze decît în una din următoarele situații:

— nu are soluții posibile;
— are soluții posibile dar optimul este infinit;
— are soluție optimă finită;

este ușor de verificat că orice problemă (T) admite soluție optimă finită. Aceasta deoarece $x_{ij} \leqslant \min(a_i, b_j)$ și deci situația optimului infinit se exclude.

Putem imediat, retranscrie problema (T), cu notațiile matriceale obișnuite unui program liniar general.

Punînd X pentru vectorul coloană al necunoscutelor problemei, de componente $x_{11}, x_{12}, \ldots, x_{1n}, x_{21}, x_{22}, \ldots, x_{2n}, \ldots, x_{m1}, x_{m2}, \ldots, x_{mn}$, b pentru vectorul coloană Q de compo-

nente $a_1, a_2, \ldots, a_m, b_1, b_2, \ldots, b_n$, şi $C = (c_{11}, c_{12}, \ldots, c_{1n}, c_{21}, c_{22}, \ldots, c_{2n}, \ldots, c_{m1}, c_{m2}, \ldots, c_{mn})$, problema (T) se reformulează astfel:

$$AX = b,$$

$$X \geqslant O,$$

$$\min CX$$

unde A este matricea de ordin $(m + n) \times mn$

$$A = \left\| \begin{matrix} I & O & \ldots & O \\ O & I & \ldots & O \\ \ldots & \ldots & \ldots & \ldots \\ O & O & & I \\ E_n & E_n & \ldots & E_n \end{matrix} \right\} \begin{matrix} \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \\ \\ \\ \\ m \text{ linii} \end{matrix}$$

(I este aici vectorul linie $(1, 1, \ldots, 1)$ cu n componente, O este vectorul nul $(0, 0, \ldots, 0)$ cu n componente, iar E_n matricea unitate de ordin n).

Este uşor de verificat că rangul matricei A este $n + m - 1$.

Ca atare, soluţiile de bază, care joacă şi în acest caz un rol central în metodologia rezolvării problemei, vor avea cel mult $n + m - 1$ componente nenule.

Deoarece justificarea teoretică a procedeelor de rezolvare a problemei transporturilor rezidă în teoremele de bază ale programării liniare (§ 4), ne vom mulţumi, în cele ce urmează, să prezentăm numai descrierea acestor procedee.

8.2. Determinarea soluţiei iniţiale de bază

Particularităţile problemei (T), pe lîngă faptul că dau o garanţie a existenţei soluţiilor de bază, creează şi posibilităţile elaborării unor procedee mai simple, în comparaţie cu cele uzitate într-un program liniar general, pentru determinarea acestora.

Scanned with OKEN Scanner

Vom expune două asemenea procedee:

I. Metoda diagonalei. (metoda colțului nord-vest)

Datele problemei (T), respectiv, cantitățile disponibile a_i la sursele A_i $(i = 1, 2, \ldots, m)$ și cererile b_j corespunzătoare destinațiilor B_j, $(j = 1, 2, \ldots, n)$ se dispun pe laturile unui tabel dreptunghiular (tabelul 19), celulele din interiorul acestora rezervîndu-se pentru cele nm necunoscute x_{ij}, ce trebuie determinate.

Tabelul 19

		...		...		a_1
		...		...		a_2
...	...	...	...	...	...	⋮
		...		...		a_i
...	...	...	...	...	...	⋮
		...		...	...	a_m
b_1	b_2	...	b_j	...	b_n	S

Componentele bazice ale soluției de bază vor fi determinate treptat începînd cu x_{11} (din colțul stîng al părții superioare a tabloului) în felul următor:

Se alege $x_{11} = \min(a_1, b_1)$.

În același timp vor fi considerate nebazice, și deci, li se va atribui valoarea 0, toate variabilele de pe aceeași linie cu x_{11} sau toate variabilele situate pe aceeași coloană cu x_{11}, în conformitate cu una din cele trei alternative ce pot avea loc:

i) dacă $a_1 < b_1$, atunci se pune $x_{11} = a_1$, iar $x_{1j} = 0$ $(j = 2, 3, \ldots, n)$ sînt componente nebazice;

ii) dacă $a_1 > b_1$, atunci se pune $x_{11} = b_1$ și $x_{i1} = 0$ $(i = 2, 3, \ldots, m)$, cu specificația că x_{i1} sînt componente nebazice;

Scanned with OKEN Scanner

iii) dacă $a_1 = b_1$, atunci se pune $x_{11} = a_1 = b_1$ şi la alegere $x_{12} = 0$ sau $x_{21} = 0$, toate celelalte componente de pe linia 1 şi coloana 1 fiind considerate nebazice, şi, implicit, nule.

Concomitent, se modifică valorile lui a_1 şi b_1 înlocuindu-le cu $a_1 - x_{11}$, $b_1 - x_{11}$.

Procedeul se repetă la pasul următor pentru celulele rămase necompletate care formează un tablou cu $m - 1$ şi n coloane; m linii şi $n - 1$ coloane; sau $m - 1$ linii şi $n - 1$ coloane, după cum primul pas s-a desfăşurat în condiţiile i), ii), respectiv iii). Prima componentă ce va fi determinată acum va ocupa aceeaşi poziţie (colţul stînga-sus), în noul tabel.

Procesul continuă terminîndu-se în cel mult $m + n - 1$ paşi, la fiecare pas determinîndu-se complet o linie (situaţia i)) o coloană (situaţia ii)), sau şi o linie şi o coloană (situaţia iii)).

Se remarcă că la fiecare pas, se determină o singură componentă nenulă.

Observaţie. De regulă, componentele nebazice nu se completează în tabel, celulele corespunzătoare rămînînd libere. Aceasta pentru a nu se confunda componentele nebazice cu eventualele componente bazice nule.

Exemplul 1.

$a_1 = 70,$ $a_2 = 10,$ $a_3 = 20;$

$b_1 = 50,$ $b_2 = 25,$ $b_3 = 15,$ $b_4 = 10.$

Condiţiile (5′) fiind verificate, considerăm o problemă de tipul (T), pentru care vom căuta o soluţie de bază cu $m + n - 1 = 6$ componente bazice (numărul total al variabilelor este $3 \times 4 = 12$).

Tabelul 20

50	20			70, 20, 0
	5	5		10, 5, 0
		10	10	20, 10
50 0	25 5 0	15 10 0	10	

Scanned with OKEN Scanner

Primul pas: întrucît $\min(a_1, b_1) = b_1 = 50$, punem, conform ii):

$x_{11} = 50$, $x_{21} = x_{31} = 0$ (am haşurat celulele corespunzătoare variabilelor nebazice x_{21}, x_{31}).

Pasul al doilea: se pleacă de la tabelul format prin înlăturarea primei coloane; de asemenea s-a modificat valoarea lui a_1 ($a_1' = a_1 - x_{11} = 20$). Deoarece $\min(a_1', b_2) = a_1' = 20$, punem (conform i)), $x_{12} = 20$, $x_{13} = x_{14} = 0$, (x_{13} şi x_{14} nebazice).

Pasul al treilea: a rămas necompletat tabelul format din liniile a doua şi a treia şi ultimele trei coloane. S-a modificat $b_2 (b_2' = b_2 - x_{12} = 5)$.

Cum $\min(a_2, b_2') = b_2' = 5$, punem $x_{22} = 5$, $x_{32} = 0$.

Pasul al patrulea: cu a_2 modificat ($a_2' = a_2 - x_{22} = 5$). Avem $\min(a_2', b_3) = a_2' = 5$, deci $x_{23} = 5$, $x_{24} = 0$. Pasul al cincilea: obligatoriu $x_{33} = 10$, $x_{34} = 10$. S-a obţinut o soluţie de bază nedegenerată

$$x_{11} = 50,\ x_{12} = 20,\ x_{22} = 5,\ x_{23} = 5,\ x_{33} = 10,\ x_{34} = 10,$$

$$x_{13} = x_{14} = x_{21} = x_{24} = x_{31} = x_{32} = 0.$$

Exemplul 2.

$$a_1 = 3,\ a_2 = 4,\ a_3 = 7;$$

$$b_1 = 1,\ b_2 = 3,\ b_3 = 3,\ b_4 = 2,\ b_5 = 5.$$

Tabelul 21

1	2				3. 2. 0
	1	3			4. 3. 0
		0	2	5	7. 5
1 0	3 1 0	3 0	2 0	5	

La pasul al patrulea, pentru tabelul format din ultimele două linii şi ultimele trei coloane, am avut $a_2' = 3$, $b_3 = 3$ şi $\min(a_2', b_3) = a_2' = b_3 = 3$.

Scanned with OKEN Scanner

Conform iii), am pus $x_{23} = 3$, $x_{33} = 0$ (componentă bazică), $x_{24} = x_{25} = 0$ (componente nebazice).

În final soluția obținută este degenerată, numărul componentelor nenule fiind 6 deci inferior lui $m + n - 1 = 7$.

II. *Metoda costurilor minime*

În determinarea soluției de bază se iau în considerare și costurile c_{ij}.

Față de procedeul precedent diferă doar ordinea de alegere a componentelor ce se determină la fiecare pas. Astfel la primul pas se determină componenta x_{kl} pentru care $c_{kl} = \min c_{ij}$.

Se pune $x_{kl} = \min (a_k, b_l)$. Aceleași trei alternative i), ii), iii) pot avea loc, după care, în al doilea pas, se repetă procedeul pentru tabelul redus, urmărindu-se minimul costurilor corespunzătoare celulelor necompletate.

Exemplul 3. Datele din exemplul 1 le completăm cu matricea costurilor c_{ij}. Pentru comoditate, acestea au fost înscrise în colțul fiecărei celule (tabelul 22).

Tabelul 22

5 / 50	6 / 5	2 / 5	3 / 10	70 65. 55. 5. 0
2 / 2	2 / 2	1 / 10	4 / 4	10. 0
6 / 6	8 / 20	3 / 3	4 / 4	20
50 0 0	25 20	15 5 0	10 0	

Primul pas: $\min\limits_{\substack{1 \leqslant i \leqslant 3 \\ 1 \leqslant j \leqslant 4}} c_{ij} = c_{23} = 1$. Ca atare punem

$x_{23} = \min (a_2, b_3) = a_2 = 10$, $x_{21} = x_{22} = x_{24} = 0$ (componente nebazice).

Scanned with OKEN Scanner

Pasul al doilea: după completarea liniei a doua, şi modificarea lui $b_3 (b_3' = b_3 - x_{23} = 5)$, căutăm

$$\min_{\substack{i=1,3 \\ 1 \leqslant j \leqslant 4}} c_{ij} = c_{13} = 2. \text{ Deci } x_{13} = \min (a_1, b_3') = b_3' = 5, \; x_{33} = 0$$

Pasul al treilea: $\min\limits_{\substack{i=1,3 \\ j=1,2,4}} c_{ij} = c_{14} = 3$. Avem $x_{14} = \min (a_1', b_4) = b_4 = 10$, $x_{24} = x_{34} = 10$.

Pasul al patrulea: după modificarea $a_1'' = a_1' - a_{14} = 55$, avem $\min\limits_{\substack{i=1,3 \\ j=1,2}} c_{ij} = c_{11} = 5$, şi deci $x_{11} = \min(a_1'', b_1) = b_1 = 50$, $x_{21} = x_{31} = 0$

Pasul al cincilea: $x_{12} = 5$, $x_{32} = 20$; obligatoriu pentru satisfacerea restricţiilor problemei.

Metoda costurilor minime, dă adesea o soluţie iniţială de bază mai bună decît metoda diagonalei, în sensul că realizează o valoare a cheltuielilor de transport mai mică. Drept urmare, numărul iteraţilor ulterioare, necesare pentru atingerea optimului, poate fi mai mic.

8.3. Determinarea soluţiei optime. Descrierea algoritmului

Pentru optimizarea soluţiei se utilizează algoritme speciale, care constituie adaptări ale metodei simplex la condiţiile particulare ale problemei (T).

Un astfel de algoritm, care va fi expus în continuare, utilizează o observaţie importantă ce decurge din forma specială a matricei A.

Notînd cu P_{kl} vectorul coloană $n + m$-dimensional al matricei A, corespunzător variabilei x_{kl}, $(k = 1, 2, \ldots, m;\; l = 1, 2, \ldots, n)$, să presupunem că dispunem de o bază B formată din $n + m - 1$ vectori P_{kl} liniar independenţi.

Dată fiind forma particulară a matricei A se poate deduce că orice vector nebazic P_{ij}, se poate reprezenta în baza B astfel:

$$P_{ij} = \sum_{P_{kl} \in B} z_{kl}^{ij} P_{kl} \tag{6}$$

unde z_{kl}^{ij} pot fi -1, 0 sau $+1$.

Scanned with OKEN Scanner

Deci, componentele lui Z_{ij} (expresia lui P_{ij} în baza B) nu vor putea avea decît cele trei valori specificate.

De aici un test pentru optimalitatea unei soluții de bază.

Deoarece problema (T) este un program liniar de minim, conform teoriei generale, o soluție este optimă dacă toți $c_{ij} - z_{ij} \geqslant 0$, adică,

$$(7) \quad c_{ij} - C_B Z_{ij} \geqslant 0, \quad \text{pentru } i = 1, 2, \ldots, m; \quad j = 1,2, \ldots, n$$

Ținînd seama de observația precedentă, membrul stîng din (7) este suma algebrică a costului c_{ij} cu o parte din costurile asociate componentelor bazice ale soluției.

Dacă condiția (7) nu este îndeplinită rezultă că soluția poate fi îmbunătățită. Pentru aceasta se modifică baza B, introducînd un vector nebazic P_{ij} în locul unuia bazic.

Vectorul P_{ij} poate fi oricare pentru care $c_{ij} - z_{ij} < 0$, recomandîndu-se de regulă pe cel corespunzător diferenței minime.

Modificarea corespunzătoare a soluției se va face atribuind componentei x_{ij} o valoare nenulă θ.

Mărimea acesteia se determină provocînd un transport al lui θ în ciclul componentelor x_{kl} corespunzător coeficienților nenuli din (6) adică scăzînd și adunînd succesiv această valoare din componentele bazice, de așa manieră încît suma totală rezultantă să rămînă invariată (într-adevăr, este ușor de observat că pentru orice soluție posibilă X a problemei $\sum_{i,j} x_{ij} = S$). Pentru că noua soluție trebuie să rămînă o soluție posibilă, deci să aibă componente nenegative, lui x_{ij} i se poate atribui valoarea maximă a lui θ pentru care toate diferențele $x_{kl} - \theta$ rămîn nenegative.

Procesul se repetă, avînd drept rezultat un șir finit de iterate, pînă la verificarea testului de optimalitate.

Vom prezenta explicit algoritmul descris pe un exemplu numeric.

Exemplul 4. Fie problema de tip (T), cu datele din tabelul 23.

Scanned with OKEN Scanner

Tabelul 23

Sursa \ Destinația	1	2	3	4	
1	3	2	2	4	70
2	1	2	3	4	10
3	3	2	2	1	20
	50	25	15	10	

În interiorul tabelului 23 au fost trecute costurile unitare de transport.

Putem porni cu soluția inițială de bază, determinată în exemplul 1 (tabelul 23):

Tabelul 24

	3 50	2 20	2	4
(S_1)	1	2 5	3 5	4
	3	2	2 10	1 10

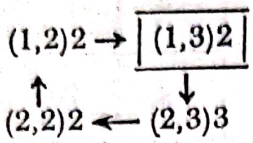
$f = 245.$

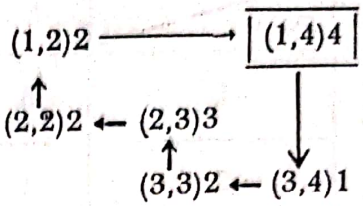
Am trecut în fiecare celulă, în colț, costul corespunzător.

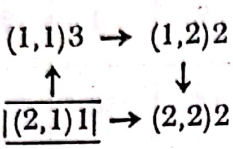
Asociem fiecărei celule libere (corespunzătoare componentelor nebazice) un ciclu care începe și se termină în celula respectivă, și ale cărui noduri sînt obligatoriu celule corespunzătoare componentelor bazice, (vom nota (i, j) pentru celula corespunzătoare variabilei x_{ij}; alăturat vom pune valoarea costului c_{ij}).

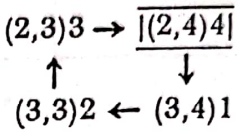
Scanned with OKEN Scanner

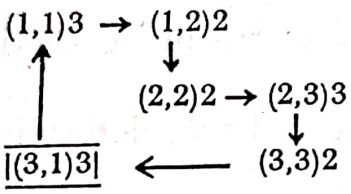
Avem următoarele cicluri

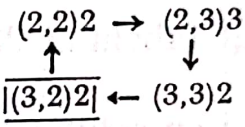












Pentru verificarea optimalității soluției alcătuim un tabel ajutător, în care trecem în celula lui x_{ij}, cantitatea $\delta_{ij} = c_{ij} - z_{ij}$ egală cu suma algebrică, cu semn alternat, a costurilor ce apar în ciclul lui (i, j), în care se începe cu $+ c_{ij}$.

Astfel:

$$\delta_{13} = 2 - 3 + 2 - 2 = -1,$$

$$\delta_{14} = 4 - 1 + 2 - 3 + 2 - 2 = 2,$$

$$\delta_{21} = 1 - 3 + 2 - 2 = -2,$$

$$\delta_{24} = 4 - 1 + 2 - 3 = 2,$$

$$\delta_{31} = 3 - 3 + 2 - 2 + 3 - 2 = 1,$$

$$\delta_{32} = 2 - 2 + 3 - 2 = 1.$$

Scanned with OKEN Scanner

Cu aceste date completăm tabelul 25.

Tabelul 25

(Δ_1)

		−1	2
−2			2
1	1		

Deoarece apar diferenţe $\delta_{ij} = c_{ij} - z_{ij}$ negative, soluţia (S_1) nu este optimă.

Prima iterată

Corespunzător celei mai mici diferenţe negative (−2), introducem în baza vectorul P_{21}.

Ca atare în ciclul corespunzător celulei (2,1) punînd în celule vechile componente bazice, adunăm şi scădem θ, pînă la echilibrarea sumei totale

$$\begin{array}{ccc} 50 - \theta & \rightarrow & 20 + \theta \\ \downarrow & & \downarrow \\ \theta & \leftarrow & 5 - \theta \end{array}$$

Valoarea maximă ce i se poate atribui lui θ este 5 (în cazul cînd θ excede 5, cel puţin una din componente devine negativă).

Deci o nouă soluţie se va obţine, cu $x_{21} = 5$ şi $x_{22} = 0$ (nebazică, vectorul P_{22} ieşind din bază).

Tabelul 26

(S_2)

45 | 3	25 | 2	| 2	| 4
5 | 1	| 2	5 | 3	| 4
| 3	| 2	10 | 2	10 | 1

, $f = 235$,

Scanned with OKEN Scanner

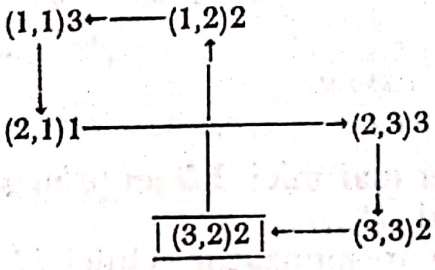
Pentru testarea optimalității noii soluții, formăm cu același procedeu ilustrat anterior, tabelul ajutător 27.

Tabelul 27

(Δ_2)

		−3	0
	2		
3	3		

De exemplu, ciclul lui (3, 2) a fost:



A doua iterată

Precedenta soluție nefiind optimă, se introduce în bază P_{13}, corespunzător diferenței $\delta_{13} = c_{13} - z_{13} = -3$.

Pentru determinarea valorii lui x_{13}, se caută θ, din ciclul

$$\begin{array}{ccc} 45 - \theta & \longrightarrow & \theta \\ \uparrow & & \downarrow \\ 5 + \theta & \leftarrow & 5 - \theta. \end{array}$$

Clar, $\theta = 5$, și ca atare vom avea o nouă soluție (tabelul 28).

Tabelul 28

(S_2)

40 | 3	25 | 2	5 | 2	4 | 4
10 | 1	| 2	| 3	| 4
| 3	| 2	10 | 2	10 | 1

, $f = 220$.

Calculăm diferenţele δ_{ij} (tabelul 29).

Tabelul 29

				3
(Δ_3)		2	3	5
	0	0		

Testul de optimalitate dă răspuns afirmativ, deoarece toate $\delta_{ij} \geqslant 0$. Soluţia optimă a problemei este deci

$$x_{11}^* = 40, \quad x_{12}^* = 25, \quad x_{13}^* = 5, \quad x_{21}^* = 1, \quad x_{33}^* = 10, \quad x_{34}^* = 10,$$

$$x_{14}^* = x_{22}^* = x_{23}^* = x_{24}^* = x_{31}^* = x_{32}^* = 0.$$

Costul total de transport asociat acestei soluţii este $f(X^*) = 220$.

8.4. Ciclaje. Tehnica perturbării

Ca şi în cazul general, şi pentru o problemă de tip (T) soluţiile degenerate pot creea condiţiile producerii fenomenului de ciclaj.

Pentru evitarea acestei eventualităţi se propune utilizarea tehnicii perturbării.

În situaţiile în care apar premizele unei soluţii degenerate se reformulează problema punînd:

$$a_i(\varepsilon) = a_i + \varepsilon, \quad i = 1, \ 2, \ \ldots \ , m;$$

$$b_n(\varepsilon) = b_n + m\varepsilon, \quad b_n(\varepsilon) = b_j, \quad j = 1. \ 2, \ \ldots \ n - 1.$$

ε fiind considerat pozitiv arbitrar de mic, se rezolvă problema în raport cu datele $a_i(\varepsilon)$, $b_j(\varepsilon)$, în final punînd $\varepsilon = 0$.

Scopul acestei tehnici, este de a determina componentele bazice nule în opoziţie cu cele nebazice, de asemenea nule.

Exemplul 5. Reluăm problema din exemplul 2. După cum am văzut, acesta produce o soluţie degenerată.

Scanned with OKEN Scanner

Folosind tehnica perturbării, căutăm o soluție de bază pentru problema corespunzînd datelor:

$$a_1(\varepsilon) = 3 + \varepsilon, \quad a_2(\varepsilon) = 4 + \varepsilon, \quad a_3(\varepsilon) = 7 + \varepsilon;$$

$$b_1(\varepsilon) = 1, \; b_2(\varepsilon) = 3, \; b_3(\varepsilon) = 3, \; b_4(\varepsilon) = 2, \; b_5(\varepsilon) = 5 + 3\varepsilon.$$

Tabelul 30

1	$2 + \varepsilon$				$3 + \varepsilon,\ 2 + \varepsilon\ 0$
	$1 - \varepsilon$	3	2ε		$4 + \varepsilon,\ 3 + 2\varepsilon,\ 2\varepsilon,\ 0$
			$2 - 2\varepsilon$	$5 + 3\varepsilon$	$7 + \varepsilon$
1 0	3 $1 - \varepsilon$ 0	3 0	2 $2 - 2\varepsilon$	$5 + 3\varepsilon$	

În final, punînd $\varepsilon = 0$, obținem soluția problemei date (tabelul 31).

Tabelul 31

1	2			
	1	3	0	
			2	5

Capitolul III

PROGRAMAREA NELINIARĂ

§ 1. CAZUL GENERAL

1.1. Un exemplu de program neliniar atașat unei probleme de stocare

O uzină își propune să-și creeze stocuri pentru n materii prime $M_1, M_2, \ldots, M_n$, de primă importanță în procesele de producție proprii.

În abordarea problemei se au în vedere următoarele date:

— costul unei unități din resursa M_i este $s_i (i = 1, 2, \ldots, n)$;

— uzina poate să aloce lunar, pentru scopul propus o sumă maximă S;

— necesarul, pentru consumul intern, din resursa M_i este aleator urmînd — după estimările făcute — o lege normală cu valoarea medie m_i și abaterea medie pătratică $\sigma_i (i = 1, 2, \ldots, \ldots, n)$;

— absența în stoc a unei unități din resursa M_i induce, pînă la procurarea sa, datorită întîrzierilor în procesul de producție, o pierdere egală cu $p_i (i = 1, 2, \ldots, n)$;

— excesul de resurse în stoc aduce prejudicii materiale evaluate la q_i pentru unitatea de resursă $M_i (i = 1, 2, \ldots, n)$;

— prin prelucrarea unei unități din resursa M_i, uzina realizează un beneficiu $r_i (i = 1, 2, \ldots, n)$

Ținînd seama de considerentele expuse, să se determine cantitățile optime $x_1, x_2, \ldots, x_n$ din resursele $M_1, M_2, \ldots, M_n$ ce trebuie stocate lunar pentru a asigura uzinei un beneficiu maxim (în evaluarea beneficiului se consideră factorii activi, doar resursele $M_1, M_2, \ldots, M_n$).

Scanned with OKEN Scanner

Din analiza datelor problemei se desprinde imediat necesitatea impunerii unor restricții asupra cantităților $x_1, x_2, \ldots, x_n$. Acestea vor fi

$$(1) \qquad \begin{aligned} & s_1x_1 + s_2x_2 + \ldots + s_nx_n \leqslant S, \\ & x_1 \geqslant 0,\ x_2 \geqslant 0, \ldots, x_n \geqslant 0. \end{aligned}$$

Prima inegalitate exprimă condiția ca valoarea resurselor stocate să nu depășească suma totală alocată acestui scop. Celelalte inegalități, privind pozitivitatea cantităților stocate sînt naturale.

În calcularea beneficiului R, obținut de uzină prin prelucrarea resurselor stocate, trebuie să se aibă în vedere faptul că necesitățile lunare $c_i (i = 1, 2, \ldots, n)$, din cele n resurse sînt variabile aleatoare.

Ca atare modelul matematic va trebui să se refere la beneficiul mediu lunar.

Evident, $R = \sum_{i=1}^{n} R_i$, unde $R_i (i = 1, 2, \ldots, n)$ reprezintă beneficiul mediu datorat prelucrării resursei M_i. Pentru evaluarea lui R_i vom face o discuție asupra cererii:

Fie $n_i(c_i ; m_i, \sigma_i)$ densitatea de probabilitate a legii normale $N(m_i, \sigma_i)$, care descrie comportarea necesarului c_i din resursa M_i în procesul de producție.

Dacă $c_i < x_i$, atunci se realizează beneficiul r_ic_i.

Dacă $c_i \geqslant x_i$, atunci uzina nu poate lucra cu întreaga capacitate realizînd un beneficiu r_ix_i.

Ca atare, în medie, beneficiul rezultat din prelucrarea resursei M_i va fi

$$r_i \int_{-\infty}^{x_i} c_i n(c_i ;\ m_i, \sigma_i) dc_i + r_i x_i \int_{x_i}^{+\infty} n(c_i ; m_i, \sigma_i)\ dc_i.$$

Totodată, uzina poate suferi pierderi datorate neconcordanței necesarului cu stocul, anume:

Scanned with OKEN Scanner

Dacă $c_i < x_i$, se suportă penalităţile datorate excesului de stoc, egale în medie cu

$$q_i \int_{-\infty}^{x_i} (x_i - c_i) n(c_i ;\ m_i,\ \sigma_i)\ \mathrm{d}c_i.$$

Dacă $c_i \geqslant x_i$, pierderile datorate absenţei materiei prime sînt în medie, egale cu

$$p_i \int_{x_i}^{+\infty} (c_i - x_i) n(c_i ;\ m_i ;\ \sigma_i) \mathrm{d}c_i.$$

Ca urmare, beneficiul net mediu datorat prelucrării stocului M_i va fi

$$R_i = r_i \int_{-\infty}^{x_i} c_i n(c_i ;\ m_i,\ \sigma_i)\mathrm{d}c_i + r_i x_i \int_{x_i}^{+\infty} n(c_i ;\ m_i,\ \sigma_i)\mathrm{d}c_i -$$

$$- q_i \int_{-\infty}^{x_i} (x_i - c_i) n(c_i ;\ m_i,\ \sigma_i)\mathrm{d}c_i - p_i \int_{x_i}^{+\infty} (c_i - x_i) n(c_i,\ m_i,\ \sigma_i)\mathrm{d}c_i =$$

$$= m_i(r_i + q_i) - q_i x_i + (r_i + q_i + p_i) \int_{x_i}^{+\infty} (x_i - c_i) n(c_i ;\ m_i,\ \sigma_i)\mathrm{d}c_i.$$

La această ultimă expresie am ajuns ţinînd seama de faptul că

$$\int_{-\infty}^{+\infty} c_i n(c_i ;\ m_i,\ \sigma_i)\ \mathrm{d}c_i = m_i,$$

şi deci

$$\int_{-\infty}^{x_i} c_i n(c_i ;\ m_i,\ \sigma_i)\ dc_i = m_i - \int_{x_i}^{+\infty} c_i n(c_i ;\ m_i,\ \sigma_i)\mathrm{d}c_i$$

Scanned with OKEN Scanner

și analog

$$\int_{-\infty}^{+\infty} (x_i - c_i)n(c_i\,;\, m_i,\, \sigma_i)\mathrm{d}c_i = x_i - m_i$$

de unde

$$\int_{-\infty}^{x_i} (x_i - c_i)n(c_i\,;\, m_i,\, \sigma_i)\mathrm{d}c_i = x_i - m_i - \int_{x_i}^{+\infty} (x_i - c_i)n(c_i\,;\, m_i,\, \sigma_i)\mathrm{d}c_i.$$

Atunci:

$$R = \sum_{i=1}^{n} m_i(r_i + q_i) - \sum_{i=1}^{n} q_i x_i +$$

$$+ \sum_{i=1}^{n} (r_i + q_i + p_i) \int_{x_i}^{+\infty} (x_i - c_i)n(c_i\,;\, m_i,\, \sigma_i)\mathrm{d}c_i.$$

Modelul matematic atașat problemei propuse cere determinarea necunoscutelor $x_1, x_2, \ldots, x_n$ care, în condițiile (1) să satisfacă:

(2) maxim R.

Este evident, că acest model se încadrează în limitele modelelor generale ale programării matematice.

Tot atît de evident este și faptul că R nu este o funcție liniară de variabilele $x_1, x_2, \ldots, x_n$.

Problema propusă aici, aparține clasei problemelor de programare neliniară.

1.2. Cîteva considerații asupra specificului problemelor de programare neliniară

Problemele de programare neliniară s-au impus studiului matematicienilor în același timp cu cele ale programării liniare, fiind reclamate de o gamă largă de modele de natură diversă.

După cum indică denumirea, o problemă de programare neliniară conține în expresia sa matematică funcții de un tip mult

Scanned with OKEN Scanner

mai general decît cele ce apar în cazul liniar; mai precis, dacă problema (pe care o vom numi uneori, pe scurt, program) se poate enunța prin

$$(1) \qquad g_i(x_1, x_2, \ldots, x_n)\{\geqslant, =, \leqslant\}\, 0, \quad i = 1, 2, \ldots, n;$$

$$(2) \qquad \text{maxim (minim)}\ f(x_1, x_2, \ldots, x_n),$$

f și g_i $(i = 1, 2, \ldots, n)$, sau o parte din acestea sînt funcții neliniare de variabilele reale $x_1, x_2, \ldots, x_n$.

Această caracteristică a problemei ridică dificultăți substanțiale de ordin matematic în studiul proprietăților generale ale soluțiilor și în elaborarea tehnicilor de calcul ale acestora.

Prima lucrare teoretică fundamentală în domeniul programării neliniare, care a stat la baza majorității cercetărilor ulterioare în acest domeniu, a fost rodul cercetărilor lui H. W. Kuhn și A. W. Tucker în direcția stabilirii unor condiții necesare și suficiente pentru existența soluțiilor și a fost publicată în anul 1951.

Deși, pînă în prezent, nu s-a reușit să se obțină o metodă eficientă de rezolvare a problemelor de programare neliniară, în forma lor cea mai generală, au fost elaborate metodologii suficient de convenabile pentru diferite forme particulare ale problemei, în special pentru tipurile de programe convexe. Prima metodă eficientă de calcul a soluțiilor pentru un tip particular de problemă (cu restricții liniare) a fost dată în anul 1960 de J. B. Rosen.

Pot fi citați și alți cercetători care, înainte de aceasta sau ulterior, au elaborat diferite algoritme de calcul, sau care au aprofundat aspectele teoretice generale ale problemelor programării neliniare: P. Wolfe, E. M. L. Beale, E. Barankin, G. Zoutendijk, H. Theil, C. Van de Panne, R. Dorfman, K. Arrow, L. Hurwicz R. A. K. Frisch, D. A. d'Esopo etc.

De notat că în stabilirea modelelor matematice ale fenomenelor concrete care conduc la probleme de tipul programării neliniare un rol important revine metodelor probabilistice, funcțiile economice fiind în general deduse în urma unei cercetări statistice a fenomenului.

În continuare vom căuta pe exemple foarte simple să punem în evidență cîteva caracteristici ale unui program neliniar care

Scanned with OKEN Scanner

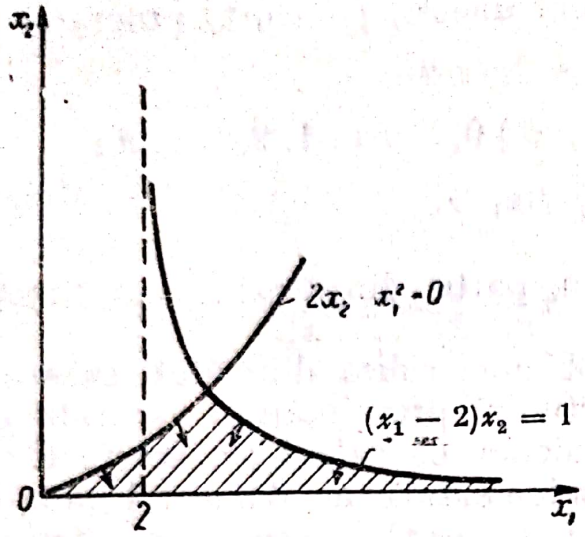


Fig. 10.

fac inaplicabilă teoria matematică elaborată în cazul programării liniare.

Exemplul 1. Fie problema în care restricțiile sînt

$$\text{I } 2x_2 - x_1^2 \leqslant 0,$$

$$(x_1 - 2)x_2 \leqslant 1;$$

$$\text{II } x_1 \geqslant 0,\ x_2 \geqslant 0.$$

Grafic față de un sistem de axe rectangulare mulțimea punctelor (x_1, x_2) satisfăcînd sistemul I, II este domeniul haşurat în figura 10.

Evident, acest domeniu nu este o mulțime convexă.

Datorită acestui fapt se pierde o proprietate importantă a mulțimii soluțiilor posibile — convexitatea — proprietate care, după cum am văzut, juca un rol important în fundamentarea matematică a metodei simplex de rezolvare a programelor liniare.

Exemplul 2. În condițiile

$$\text{I. } 4x_1 + 3x_2 - 12 \leqslant 0;$$

$$\text{II. } x_1 \geqslant 0,\ x_2 \geqslant 0;$$

III. Să se minimizeze funcția-obiectiv

$$f(x_1, x_2) = (x_1 - 3)^2 + (x_2 - 5)^2.$$

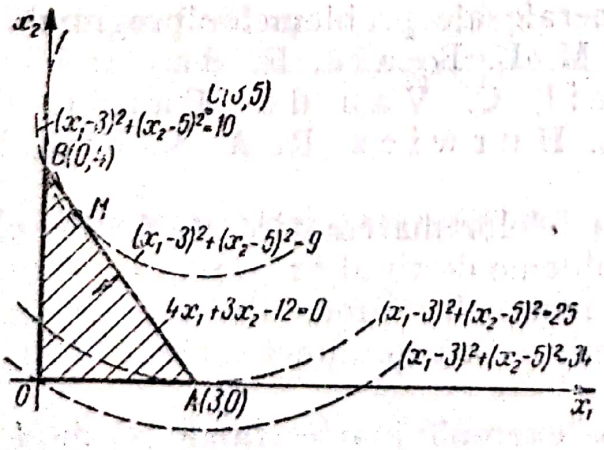


Fig. 11.

În figura 11, triunghiul OAB reprezintă mulțimea soluțiilor posibile satisfăcînd I, II.

Deși în acest caz, mulțimea susmenționată este un domeniu convex, se poate remarca că o altă proprietate importantă folo-

sită de algoritmul simplex, se pierde în cazul problemelor programării neliniare. Este vorba de faptul că, într-o astfel de problemă soluția optimă nu este, obligatoriu, atinsă într-un punct de extremum al mulțimii soluțiilor posibile (cînd aceasta este convexă).

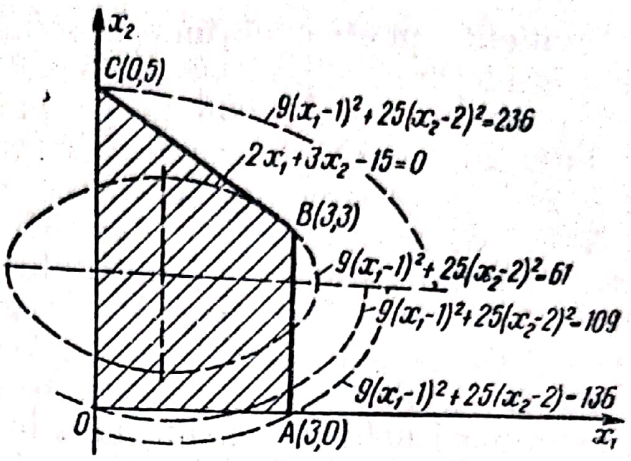


Fig. 12.

Într-adevăr, orice valoare z a funcției-obiectiv $f(x_1, x_2)$, poate fi interpretată ca pătratul razei cercului de ecuație $(x_1-3)^2+(x_2-5)^2=z$, cu centrul în punctul $C(3, 5)$.

Cum condiția III cere determinarea valorii minime z, soluția optimă a problemei va fi dată de punctul (x_1, x_2), aparținînd triunghiului OAB, care se situează pe cercul cu raza minimă.

Or, se observă că punctul M de tangență al cercului de ecuație $(x_1-3)^2+(x_2-5)^2=9$ cu segmentul AB corespunde acestui optim.

Evident, M nu este un punct extrem al mulțimii soluțiilor posibile.

Mai mult, se poate verifica că valorile lui f, corespunzătoare punctelor extreme O, A și B, — respectiv 34, 25, și 10 — sînt mai mari decît valoarea 9 atinsă în M.

Exemplul 3.

I. $x_1 \leqslant 3$,
$2x_1+3x_2-15 \leqslant 0$;

II. $x_1 \geqslant 0$, $x_2 \geqslant 0$;

III. Să se maximizeze

$$f(x_1, x_2)=9(x_1-1)^2+25(x_2-2)^2.$$

Mulțimea soluțiilor posibile este patrulaterul convex $OABC$ (figura 12).

Pentru o valoare z a funcției $f(x_1, x_2)$, egalitatea din III devine ecuația unei elipse cu axele de simetrie paralele cu axele de coordonate și cu centrul în punctul (1, 2).

Scanned with OKEN Scanner

Evident, pentru obținerea soluției optime, sîntem conduși la considerarea elipsei cu axele maxime posibile și care să aibă cel puțin un punct comun cu patrulaterul $OABC$.

Punctul astfel determinat $C(x_1 = 0,\ x_2 = 5)$, pentru care $z = 236$ reprezintă maximum căutat.

Trecînd, însă în revistă valoriile funcției obiectiv în celelalte trei puncte extreme ale poligonului soluțiilor posibile, constatăm că valoarea $z = 136$ atinsă în A este superioară valorilor $z = 109$ și $z = 61$ atașate punctelor O, respectiv B.

Acest exemplu evidențiază o caracteristică a problemelor programării neliniare, neîntîlnită în programele liniare, și anume existența în afara soluției optime globale, a unor puncte de optim local, care dau, pentru funcția obiectiv o valoare maximă (minimă) în raport cu toate punctele situate într-o anumită vecinătate.

1.3. Soluțiile problemei programării neliniare

Introducînd notația vectorială $X = (x_1, x_2, \ldots, x_n)$, problema enunțată anterior cere determinarea vectorului $X \in R^n$, satisfăcînd restricțiile

$$(1) \qquad g_i(X)\ \{\leqslant, =, \geqslant\}\ 0, \qquad i = 1, 2, \ldots, m,$$

și care realizează

$$(2) \qquad \text{maxim (minim) } f(X)$$

f și g_i $(i = 1, 2, \ldots, m)$ fiind funcții oarecare definite pe R^n.

Amintim două cunoscute definiții ale analizei matematice, privind extremele unei funcții:

$X^* = (x_1^*, x_2^*, \ldots, x_n^*)$ este un *punct de maxim* (minim) *local* al funcției f, dacă există o vecinătate $V_{X^*} \subset R^n$ a lui X^*, astfel ca

$$f(X) \leqslant f(X^*)\ (f(X) \geqslant f(X^*)), \text{ oricare ar fi } X \in V_{X^*}.$$

$X^* = (x_1^*, x_2^*, \ldots, x_n^*)$ *este un punct de maxim* (minim) *global* al funcției f, dacă

$$f(X) \leqslant f(X^*)\ (f(X) \geqslant f(X^*)), \text{ oricare ar fi } X \in R^n.$$

Scanned with OKEN Scanner

Pornind de la aceste noţiuni fundamentale, ajungem în mod natural la definirea soluţiilor problemei considerate.

Vom numi *spaţiul soluţiilor posibile* pentru problema de programare neliniară, submulţimea $\mathfrak{X} \subset R^n$ a tuturor vectorilor $X = (x_1, x_2, \ldots, x_n)$, care satisfac sistemul de restricţii (1). Un element $X \in \mathfrak{X}$ va fi numit *soluţie posibilă sau punct admisibil.*

O *soluţie optimă locală* a problemei de programare neliniară este un vector $X^* = (x_1^*, x_2^*, \ldots, x_n^*) \in \mathfrak{X}$ care reprezintă pentru funcţia obiectiv f un punct de maxim (minim) local.

O *soluţie optimă globală* a problemei de programare neliniară este un vector $X^* = (x_1^*, x_2^*, \ldots, x_n^*) \in \mathfrak{X}$ care reprezintă pentru funcţia obiectiv f un punct de maxim (minim) global.

Evident, rezolvarea problemei considerate, impune determinarea soluţiilor optime globale, cele locale nerăspunzînd într-o măsură suficient de bună imperativelor care au condus la formularea problemei.

1.4. Metoda lagrangeanului

Să ne limităm, pentru început, la tipul de problemă în care restricţiile au forma:

$$g_i(X) = 0, \quad i = 1, 2, \ldots, m. \tag{1'}$$

În acest caz, aflarea soluţiilor optime este echivalentă cu determinarea extremelor unei funcţii, cu legături, problemă clasică în analiza matematică. Metoda obişnuită de rezolvare pentru această problemă este aşa numita metodă a lagrangeanului.

Fie $X^* = (x_1^*, x_2^*, \ldots, x_n^*)$ o soluţie optimă (locală sau globală) a problemei (1'), (2).

Introducînd cîteva ipoteze suplimentare presupunem că f şi g_i $(i = 1, 2, \ldots, m)$, au derivate parţiale, în raport cu variabilele x_j $(j = 1, 2, \ldots, n)$, continue, $(f, g_i \in C^1)$ şi fie $D(X)$ matricea ale cărei elemente sînt $d_{ij} = \frac{\partial g_i}{\partial x_j}(X)$, $(i = 1, 2, \ldots$ $\ldots, m;\ j = 1, 2, \ldots, n)$.

Să presupunem $m < n$ şi rangul matricei $D(X^*)$ egal cu m.

Scanned with OKEN Scanner

Fără a restrînge generalitatea, se poate presupune atunci că determinantul funcţional

$$\frac{D(g_1, g_2, \ldots, g_m)}{D(x_1, x_2, \ldots, x_m)},$$

are o valoare diferită de zero în punctul $X^* = (x_1^*, x_2^*, \ldots, x_n^*)$.

Această ultimă ipoteză, face posibilă explicitarea variabilelor $x_1, x_2, \ldots, x_m$, într-o anumită vecinătate $V_{\overline{X}^*}$ a punctului $\overline{X}^* = (x_{m+1}^*, x_{m+2}^*, \ldots, x_n^*)$.

Se poate scrie deci

$$x_i = \varphi_i(\overline{X}), \text{ pentru } i = 1, 2, \ldots, m; \quad \overline{X} \in V_{\overline{X}^*}.$$

Atunci funcţia

$h(\overline{X}) = f[\varphi_1(\overline{X}), \varphi_2(\overline{X}), \ldots, \varphi_m(\overline{X}), \overline{X}]$, are un extremum în punctul X^* şi deci

$$\frac{\partial h}{\partial x_j}(\overline{X}^*) = 0 \qquad j = m+1, m+2, \ldots, n.$$

sau

(3)

$$\sum_{k=1}^{m} \frac{\partial f}{\partial x_k}(X^*) \cdot \frac{\partial \varphi_k}{\partial x_j}(\overline{X}^*) + \frac{\partial f}{\partial x_j}(X^*) = 0, \; j = m+1, \ldots, n.$$

Dacă, pentru fiecare i, $(i = 1, 2, \ldots, m)$, derivăm în raport cu x_j, $(j = m+1, \ldots, n)$, egalitatea

$$g_i(X) = 0$$

atunci

$$(4) \quad \sum_{k=1}^{m} \frac{\partial g_i}{\partial x_k}(X) \frac{\partial \varphi_k}{\partial x_j}(\overline{X}) + \frac{\partial g_i}{\partial x_j}(X) = 0, \qquad \begin{matrix} i = 1, 2, \ldots, m \\ j = m+1, \ldots, n \end{matrix}$$

Fie acum $Y = (y_1, y_2, \ldots, y_m)$ soluţia unică a sistemului liniar de ecuaţii

$$(5) \quad \sum_{i=1}^{m} y_i \frac{\partial g_i}{\partial x_k}(X^*) + \frac{\partial f}{\partial x_k}(X^*) = 0, \qquad k = 1, 2, \ldots, m.$$

Scanned with OKEN Scanner

(Existența și unicitatea soluției acestui sistem este garantată de ipoteza:

$$\frac{D(g_1, g_2, \ldots, g_m)}{D(x_1, x_2, \ldots, x_m)}(X^*) \neq 0).$$

Înmulțind cu g_i egalitățile (4) scrise în punctul X^* și sumînd după i, obținem

$$(6) \qquad \sum_{i=1}^{m} y_i \left[\sum_{i=1}^{m} \frac{\partial g_i}{\partial x_k}(X^*) \frac{\partial \varphi_k}{\partial x_j}(\overline{X}^*) \right] + \sum_{i=1}^{m} y_i \frac{\partial g_i}{\partial x_j}(X^*) = 0.$$

Adunînd (6) și (3) obținem

$$\sum_{k=1}^{m} \left[\frac{\partial f}{\partial x_k}(X^*) + \sum_{i=1}^{m} y_i \frac{\partial g_i}{\partial x_k}(X^*) \right] \frac{\partial \varphi_k}{\partial x_j}(\overline{X}^*) +$$

$$+ \frac{\partial f}{\partial x_j}(X^*) + \sum_{i=1}^{m} y_i \frac{\partial g_i}{\partial x_j}(X^*) = 0.$$

Ținînd seama de (5) rezultă de aici

$$(7) \qquad \frac{\partial f}{\partial x_j}(X^*) + \sum_{i=1}^{m} y_i \frac{\partial g_i}{\partial x_j}(X^*) = 0; \quad j = m+1, \ldots, n.$$

În definitiv din (5) și (7), tragem concluzia că, pentru ca vectorul X^* să fie o soluție optimă a programului (1'), (2), este necesar să fie satisfăcut următorul sistem

$$(8) \qquad \frac{\partial f}{\partial x_j}(X^*) + \sum_{i=1}^{m} y_i \frac{\partial g_i}{\partial x_j}(X^*) = 0, \qquad j = 1, 2, \ldots, n;$$

$$g_i(X^*) = 0 \qquad i = 1, 2, \ldots, m,$$

în care y_i se obțin prin rezolvarea sistemului de ecuații (5).

Expresia acestor condiții necesare, ne conduce la considerarea funcției lui Lagrange (lagrangean), asociată problemei de extremum,

$$(9) \qquad F(X, Y) = f(X) + \sum_{i=1}^{m} y_i g_i(X).$$

Scanned with OKEN Scanner

Aici, y_i $(i = 1, 2, \ldots, m)$ se numesc multiplicatorii lui Lagrange.

Condiţiile (8) deduse anterior se transpun uşor asupra funcţiei F, observînd că

$$\frac{\partial F}{\partial x_j} = \frac{\partial f}{\partial x_j} + \sum_{i=1}^{m} y_i \frac{\partial g_i}{\partial x_j}, \qquad j = 1, 2, \ldots, n;$$

$$\frac{\partial F}{\partial y_i} = g_i, \qquad i = 1, 2, \ldots, m.$$

Deci dacă X^* este o soluţie optimă a problemei de programare neliniară (1′) şi (2), satisfăcînd şi ipotezele suplimentare, citate anterior, atunci există un vector $Y^* = (y_1^*, y_2^*, \ldots, y_m^*) \in R^m$ astfel că

$$\begin{aligned} &\frac{\partial F}{\partial x_j}(X^*, Y^*) = 0, \qquad j = 1, 2, \ldots, n, \\ &\frac{\partial F}{\partial y_i}(X^*, Y^*) = 0, \qquad i = 1, 2, \ldots, m. \end{aligned} \tag{10}$$

Înlăturînd ipoteza restrictivă, rang $D(X^*) = m$, se poate ajunge, după o discuţie amănunţită asupra diverselor posibilităţi de soluţionare a sistemului (3), la reformularea condiţiilor (10):

$$\begin{aligned} &\frac{\partial F'}{\partial x_j}(X^*, Y^*) = 0, \qquad j = 1, 2, \ldots, n; \\ &\frac{\partial F'}{\partial y_i}(X^*, Y^*) = 0, \qquad i = 1, 2, \ldots, m, \end{aligned} \tag{11}$$

cu $F'(X, Y) = y_0 f(X) + \sum_{i=1}^{m} y_i g_i(X)$, unde y_0 poate fi 0 sau 1.

În cele ce urmează ne vom limita la cazul în care $y_0 = 1$ şi deci $F(X)$, Y are forma (9).

Scanned with OKEN Scanner

1.5. Soluția optimă și punctul șa al lagrangeanului

Fie $F: R^{n+m} \to R$. Vectorul $(X^*, Y^*) = (x_1^*, x_2^*, \ldots, x_n^*, y_1^*, y_2^*, \ldots, y_m^*)$ constituie un *punct șa local* al funcției F, dacă există o vecinătate $V_{X^*} \times V_{Y^*}$, a punctului (X^*, Y^*), astfel ca

$$F(X, Y^*) \leqslant F(X^*, Y^*) \leqslant F(X^*, Y) \tag{12}$$

pentru orice $X \in V_{X^*}$ și $Y \in V_{Y^*}$.

Vectorul (X^*, Y^*) constituie un *punct șa global* al funcției F dacă inegalitățile (12) sînt satisfăcute pentru orice $(X, Y) \in \in R^{m+n}$.

Fie acum $X^* = (x_1^*, x_2^*, \ldots, x_n^*)$ o soluție optimă (maxim) a problemei

$$g_i(X) = 0, \quad i = 1, 2, \ldots, m, \tag{1'}$$

$$\text{maxim } f(X) \tag{2}$$

și fie, $Y^* = (y_1^*, y_2^*, \ldots, y_m^*)$, sistemul multiplicatorilor lui Lagrange asociat acestui sistem (prin (5)).

Pentru simplificare vom considera că X^* este un maxim global.

Ținînd seama de (9) și (2), este clar că pentru orice $X \in \mathfrak{X}$ (punct admisibil)

$$F(X, Y) = f(X), \text{ oricare ar fi } Y = (y_1, y_2, \ldots, y_m) \tag{13}$$

Pe de altă parte

$$F(X, Y^*) \leqslant \max_{X \in \mathfrak{X}} F(X, Y^*) = \max_{X \in \mathfrak{X}} f(X). \tag{14}$$

Dar, X^* realizează maximul lui f și deci, ultimul membru din (14) este $f(X^*)$.

Din (13) și (14) deducem

$$F(X, Y^*) \leqslant F(X^*, Y^*) = F(X^*, Y) \tag{15}$$

oricare ar fi X și Y.

Observînd că relația (15) implică tocmai faptul că (X^*, Y^*) este un punct șa (degenerat) al funcției F, putem conclude: *dacă X^* este o soluție optimă (maxim) a problemei (1') (2) atunci există un vector $Y^* = (y_1^*, y_2^*, \ldots, y_m^*)$ satisfăcînd (5), astfel ca (X^*, Y^*) să fie un punct șa pentru lagrangeanul asociat problemei.*

Scanned with OKEN Scanner

Cazul în care programul cere determinarea minimului funcției obiectiv se tratează asemănător, concluzia fiind că (X^*, Y^*) este punct șa al funcției $G(Y, X) = F(X, Y)$, în sensul că

$$F(X^*, Y) \leqslant F(X^*, Y^*) \leqslant F(X, Y^*). \tag{16}$$

Din considerațiile făcute, pînă acum, asupra metodei lagrangeanului, se detașează faptul că soluției optime a programului i se asociază un sistem de parametrii reali — multiplicatorii lui Lagrange.

Cum, problema programării matematice are însă, și un conținut concret, se poate pune întrebarea dacă vectorul Y al multiplicatorilor lui Lagrange este susceptibil de a fi interpretat în termenii problemei originare.

Pe un exemplu prezentat în continuare, vom arăta cum componentele y_i $(i = 1, 2, \ldots, m)$ ale lui Y se pot plasa, în mod natural, în contextul concret al problemei, în a cărei rezolvare matematică au fost inițial introduse.

Presupunem că pentru n produse $A_1, A_2, \ldots, A_n$, o unitate economică dispune de m resurse $B_1, B_2, \ldots, B_m$, în cantitățile $b_1, b_2, \ldots, b_m$.

Dacă restricțiile generate de procesul de fabricație se exprimă matematic prin relațiile

a) $g_i(X) = b_i$, $i = 1, 2, \ldots, m$,

se pune problema determinării cantităților $x_1, x_2, \ldots, x_n$ de produse ce trebuie planificate pentru a asigura unității un beneficiu maxim.

Fie X^* o soluție optimă a problemei.

Atunci din relația a) se pot afla $x_1^*, x_2^*, \ldots, x_n^*$, ca funcții de b_i $(i = 1, 2, \ldots, m)$ și atunci beneficiul maxim $z^* = f(X^*)$, poate fi considerat, formal, funcție de $b_1, b_2, \ldots, b_m$.

Derivînd în raport cu b_i relațiile a) și expresia lui z^*, obținem

$$\frac{\partial z^*}{\partial b_i} = \sum_{j=1}^{m} \frac{\partial f}{\partial x_j^*} \frac{\partial x_j^*}{\partial b_i}, \qquad i = 1, 2, \ldots, m;$$

$$\sum_{j=1}^{m} \frac{\partial g_k}{\partial x_j^*} \frac{\partial x_j^*}{\partial b_i} = \delta_{ik} \text{ (simbolul lui Kronecker)},$$

$$k = 1, 2, \ldots, m; \; i = 1, 2, \ldots, m.$$

Scanned with OKEN Scanner

De aici, dacă $Y^* = (y_1^*, y_2^*, \ldots, y_m^*)$ este vectorul multiplicatorilor lui Lagrange asociat soluției optime X^*, se poate scrie

$$\frac{\partial z^*}{\partial b_i} = -\sum_{k=1}^{m} y_k \delta_{ik} + \sum_{j=1}^{m} \left[\frac{\partial f}{\partial x_j^*} + \sum_{k=1}^{m} y_k^* \frac{\partial g_k}{\partial x_j^*} \right] \frac{\partial x_j^*}{\partial b_i}$$

și cum din (5)

$$\frac{\partial f}{\partial x_j^*} + \sum_{k=1}^{m} y_k^* \frac{\partial g_k}{\partial x_j^*} = 0,$$

rezultă

$$\frac{\partial z^*}{\partial b_i} = -y_i^*.$$

Putem interpreta, intuitiv, ultima egalitate, atribuind lui y_i^* sensul de indicator al variației beneficiului maxim, datorat variației cantității disponibile b_i din resursa A_i.

1.6. Forma generală a restricțiilor. Restricții de semn pentru variabile

Ca și în problemele de programare liniară, sistemul de restricții al unui program neliniar poate fi, cu ușurință modificat, aducîndu-l la un sistem de egalități de tipul (1').

Acest lucru se realizează prin introducerea unor variabile de compensație.

Să presupunem că condițiile (1) sînt de forma

$$\begin{aligned} &g_i(X) \geqslant 0, && i = 1, 2, \ldots, s_1; \\ &g_i(X) \leqslant 0, && i = s_1 + 1, \ldots, s_2; \\ &g_i(X) = 0, && i = s_2 + 1, \ldots, n, \quad (s_1 < s_2 < m). \end{aligned}$$

Echivalentul acestui sistem este

$$\begin{aligned} &\overline{g}_i(X) = 0, && i = 1, 2, \ldots, m; \\ &x_{ci} \geqslant 0, && i = 1, 2, \ldots, s_2. \end{aligned}$$

Scanned with OKEN Scanner

unde

$$\bar{g}_i(X) = \begin{cases} g_i(X) - x_{ci}, & \text{pentru } i = 1, 2, \ldots, s_1; \\ g_i(X) + x_{ci}, & \text{pentru } i = s_1 + 1, \ldots, s_2; \\ g_i(X), & \text{pentru } i = s_2 + 1, \ldots, m. \end{cases}$$

În noua formă a programului, căruia i se poate aplica teoria lagrangeanului, apar însă noi condiții suplimentare de nenegativitate pentru o parte din variabile.

De altfel, asemenea condiții apar în mod natural în majoritatea problemelor de programare matematică (vezi și exemplul 1. din acest paragraf).

Să tratăm acest aspect, considerînd că problema de programare neliniară impune satisfacerea unui sistem de restricții, presupus adus la forma (1′) ceea ce, după cum am văzut nu restrînge generalitatea) și a unor condiții suplimentare asupra semnului variabililor.

Să scriem

$$\{1, 2, \ldots, n\} = J_1 \cup J_2 \cup J_3;$$

să presupunem:

$$\text{(17)} \qquad \begin{aligned} &x_j \geqslant 0, && \text{pentru } j \in J_1 \\ &x_j \leqslant 0, && \text{pentru } j \in J_2, \\ &x_j \text{ fără restricții de semn,} && \text{pentru } j \in J_3. \end{aligned}$$

Dacă programul neliniar cere realizarea maximului funcției obiectiv $f(X)$, aceasta atrage după sine, conform rezultatelor obținute anterior existența unui punct șa (X^*, Y^*) pentru funcția lui Lagrange $F(X, Y)$.

Vom căuta să deducem condițiile necesare ce trebuie satisfăcute de $F(X, Y)$ pentru ca (X^*, Y^*) să fie punct șa.

Notăm $\{1, 2, \ldots, m\} = I_1 \cup I_2 \cup I_3$,

$$\text{(18)} \qquad \begin{aligned} &y_i \geqslant 0, && \text{pentru } i \in I_1; \\ &y_i \leqslant 0, && \text{pentru } i \in I_2; \\ &y_i \text{ fără restricții de semn,} && \text{pentru } i \in I_3. \end{aligned}$$

Fie $F \in C^1$.

Scanned with OKEN Scanner

Deoarece, conform relaţiilor (12), $F(X^*, Y^*)$ realizează maximul lui $F(X, Y^*)$ în raport cu variabila X şi minimul lui $F(X^*, Y)$ în raport cu variabila Y, rezultă

$$\frac{\partial F}{\partial x_j}(X^*, Y^*) = 0, \qquad \text{pentru } j \in J_3,$$

$$\frac{\partial F}{\partial y_i}(X^*, Y^*) = 0, \qquad \text{pentru } i \in I_3.$$

Limitarea la submulţimile de indici J_3 şi I_3 se datorează faptului că celelalte componente sînt supuse restricţiilor suplimentare de semn.

Să examinăm în continuare şi aceste situaţii.

Dacă pentru un indice $j \in J_1$, $x_j^* > 0$, atunci evident, F ca funcţie de variabilă x_j, atinge în x_j^* un maxim (local sau global), şi, implicit:

$$\frac{\partial F}{\partial x_j}(X^*, Y^*) = 0.$$

Altfel, dacă $x_j^* = 0$, egalitatea de mai sus nu este obligatorie Putem remarca însă, că în acest caz, valorile lui F trebuie să descrească pe măsură ce componenta x_j se deplasează dinspre valoarea zero, şi deci, din continuitatea lui $\frac{\partial F}{\partial x_j}$:

$$\frac{\partial F}{\partial x_j}(X^*, Y^*) \leqslant 0,$$

Cele două situaţii apar, schematizate pe figura 13 a, b.

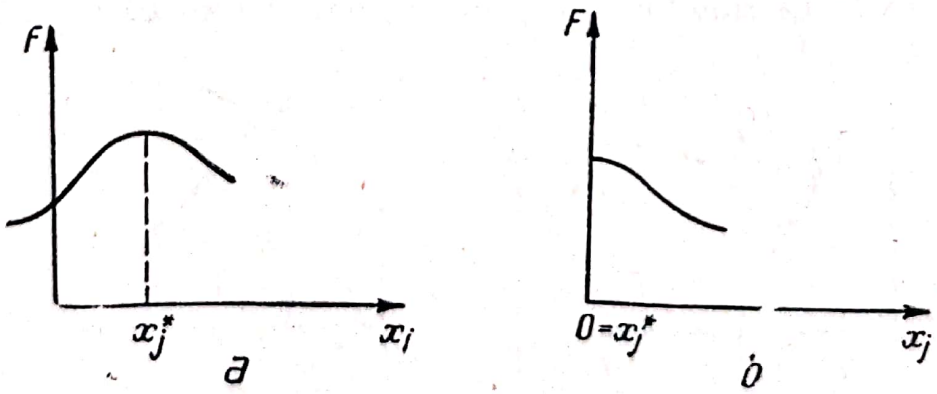


Fig. 13.

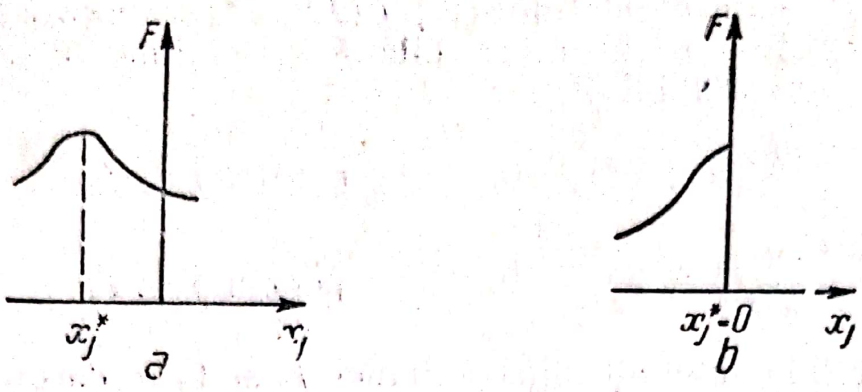


Fig. 14.

Fie acum, un indice $j \in J_2$.

Folosind aceleaşi argumente, reprezentate schematic în figura 14 a, b, deducem că

$$\frac{\partial F}{\partial x_j}(X^*, Y^*) = 0, \qquad \text{pentru } x_j^* < 0,$$

$$\frac{\partial F}{\partial x_j}(X^*, Y^*) \geqslant 0, \qquad \text{pentru } x_j^* = 0.$$

Pentru componentele y_j^* ale lui Y^*, va trebui să se ţină seama de faptul că relaţia (12) cere

$$F(X^*, Y^*) \leqslant F(X^*, Y).$$

Dacă pentru un indice $i \in I_1$, $y_i^* > 0$, evident (fig. 15, a)

$$\frac{\partial F}{\partial y_i}(X^*, Y^*) = 0.$$

Dacă însă $y_i^* = 0$, anularea derivatei nu este obligatorie, însă, valorile lui F, ca funcţie de y_i, vor trebui să crească pe măsură

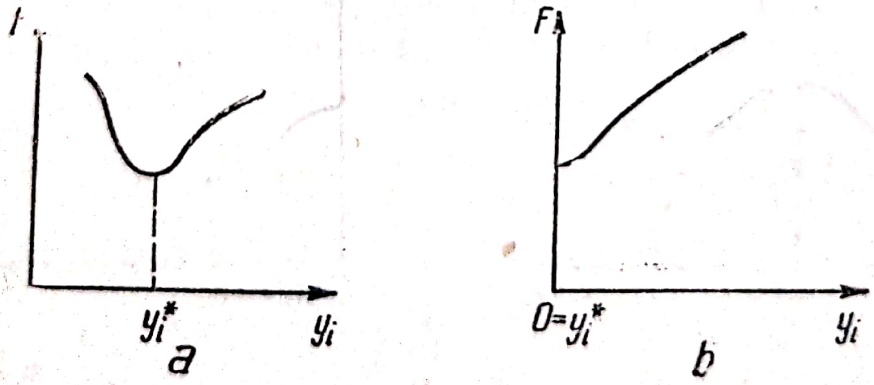


Fig. 15.

ce variabila se depărtează de valoarea de minim în sensul pozitiv al axei (fig. 15, b), și deci

$$\frac{\partial F}{\partial y_i}(X^*, Y^*) \geqslant 0.$$

Analog, dacă $i \in I_2$

$$\frac{\partial F}{\partial y_i}(X^*, Y^*) = 0, \qquad \text{pentru } y_i^* < 0 \text{ (fig. 16, a)}$$

$$\frac{\partial F}{\partial y_i}(X^*, Y^*) \leqslant 0, \qquad \text{pentru } y_i^* = 0 \text{ (fig. 16, b)}$$

În concluzie, dacă (X^*, Y^*), răspunzînd cerințelor (17) — (18), este punct șa al funcției obiectiv $F(X, Y)$ care se bucură de anumite condiții de regularități ($F \in C^1$), este necesar ca următorul sistem de condiții să fie satisfăcut:

$$\begin{aligned}
&\frac{\partial F}{\partial x_j}(X^*, Y^*) \leqslant 0, && \text{pentru } j \in J_1,\\
&\frac{\partial F}{\partial x_j}(X^*, Y^*) \geqslant 0, && \text{pentru } j \in J_2,\\
&\frac{\partial F}{\partial x_j}(X^*, Y^*) = 0, && \text{pentru } j \in J_3,\\
&x_j^* \frac{\partial F}{\partial x_j}(X^*, Y^*) = 0, && \text{pentru } j = 1, 2, \ldots, n.
\end{aligned} \tag{19}$$

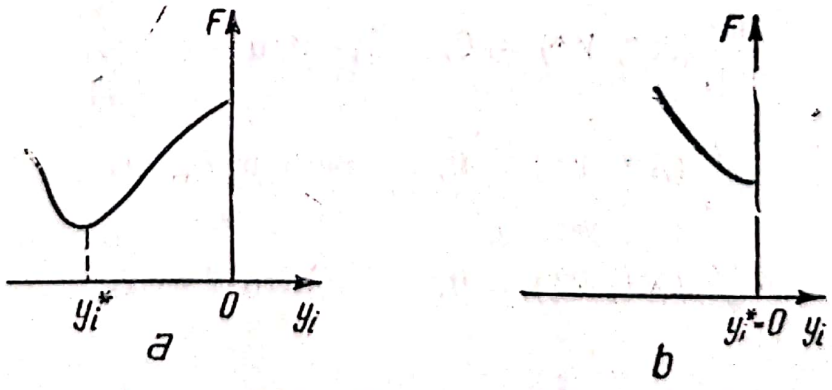


Fig. 16.

$$\frac{\partial F}{\partial y_i}(X^*, Y^*) \geqslant 0, \qquad \text{pentru } i \in I_1,$$

$$\frac{\partial F}{\partial y_i}(X^*, Y^*) \leqslant 0, \qquad \text{pentru } i \in I_2, \tag{20}$$

$$\frac{\partial F}{\partial y_i}(X^*, Y^*) = 0, \qquad \text{pentru } i \in I_3,$$

$$y_i^* \frac{\partial F}{\partial y_i}(X^*, Y^*) = 0, \qquad \text{pentru } i = 1, 2, \ldots, m.$$

În cazul programelor neliniare de minim, condițiile (19) — (20) se modifică, corespunzător interpretării relațiilor (16), ceea ce are ca efect schimbarea sensurilor inegalităților (19) și (20).

$$\frac{\partial F}{x_j}(X^*, Y^*) \geqslant 0, \qquad \text{pentru } j \in J_1,$$

$$\frac{\partial F}{\partial x_j}(X^*, Y^*) \leqslant 0, \qquad \text{pentru } j \in J_2, \tag{19'}$$

$$\frac{\partial F}{\partial x_j}(X^*, Y^*) = 0, \qquad \text{pentru } j \in J_3,$$

$$x_j^* \frac{\partial F}{\partial x_i}(X^*, Y^*) = 0, \qquad \text{pentru } j = 1, 2, \ldots, n$$

și

$$\frac{\partial F}{\partial y_i}(X^*, Y^*) \leqslant 0, \qquad \text{pentru } i \in I_1,$$

$$\frac{\partial F}{\partial y_i}(X^*, Y^*) \geqslant 0, \qquad \text{pentru } i \in I_2, \tag{20'}$$

$$\frac{\partial F}{\partial y_i}(X^*, Y^*) = 0, \qquad \text{pentru } i \in I_3,$$

$$y_i^* \frac{\partial F}{\partial y_i}(X^*, Y^*) = 0, \qquad \text{pentru } i = 1, 2, \ldots, m.$$

Scanned with OKEN Scanner

1.7. Rezolvarea numerică a unui exemplu.

Să particularizăm problema 1.1 pentru următoarele date concrete:

$n = 3$;

$S = 1\,300$; $s_1 = 5$, $s_2 = 7$, $s_3 = 10$,

$m_1 = 80$, $\sigma_1 = 20$; $m_2 = 40$, $\sigma_2 = 12$; $m_3 = 50$, $\sigma_3 = 15$;

$p_1 = 16$, $p_2 = 10$, $p_3 = 19$; $q_1 = 4$, $q_2 = 3$, $q_3 = 5$;

$r_1 = 8$, $r_2 = 6$, $r_3 = 10'$

Va rezulta deci problema:

I $5x_1 + 7x_2 + 10x_3 \leqslant 1\,300$,

$x_1 \geqslant 0$, $x_2 \geqslant 0$, $x_3 \geqslant 0$;

II $\max\{R(x_1, x_2, x_3) = 1870 - (4x_1 + 3x_2 + 5x_3) +$

$$+ 28\int_{x_1}^{+\infty}(x_1 - c_1)\cdot n(c_1, 80, 20)\mathrm{d}c_1 + 19\int_{x_2}^{+\infty}(x_2 - c_2)n(c_2; 40, 12)\mathrm{d}c_2 +$$

$$+ 34\int_{x_3}^{+\infty}(x_3 - c_3)\cdot n(c_3; 50, 15)\mathrm{d}c_3.$$

Înainte de a aplica metoda lagrangeanului, să căutăm maximul lui $R(x_1, x_2, x_3)$ fără restricţii.

Avem

$$\frac{\partial R}{\partial x_1} = -4 + 28\int_{x_1}^{\infty} n(c_1; 80, 20)\mathrm{d}c_1$$

$$\frac{\partial R}{\partial x_2} - 3 + 19\int_{x_2}^{\infty} n(c_2; 40, 12)\mathrm{d}c_2,$$

$$\frac{\partial R}{\partial x_3} = -5 + 34\int_{x_3}^{\infty} n(c_3; 50{,}1\ 5)\mathrm{d}c_3.$$

Scanned with OKEN Scanner

Dacă ţinem seama că

$$\int_x^{\infty} n(c\,;\,m,\ \sigma)\mathrm{d}c = \frac{1}{2} - \Phi\left(\frac{x-m}{\sigma}\right)$$

$\Phi(\alpha)$ fiind funcţia lui Laplace: $\int_0^{\alpha} e^{-\frac{t^2}{2}}\,dt$, obţinem prin egalarea cu zero a derivatelor parţiale, calculate anterior, sistemul de ecuaţii

III.
$$10 - 28\,\Phi\left(\frac{x_1-80}{20}\right) = 0\,;$$
$$13 - 19\,\Phi\left(\frac{x_2-40}{12}\right) = 0,$$
$$12 - 34\,\Phi\left(\frac{x_3-50}{15}\right) = 0.$$

De unde:

$$\Phi\left(\frac{x_1-80}{20}\right) \simeq 0{,}35071,\quad \Phi\left(\frac{x_2-40}{12}\right) \simeq 0{,}34210,\quad \Phi\left(\frac{x_3-50}{15}\right) \simeq 0{,}35294.$$

Scoţînd valorile argumentelor din tabelele funcţiei ϕ, găsim soluţia $x_1 \simeq 101$, $x_2 \simeq 52$, $x_3 \simeq 66$.

Valorile găsite satisfac condiţiile de pozitivitate dar nu şi inegalitatea I. De aici necesitatea folosirii acesteia ca o restricţie activă.

Vom considera atunci lagrangeanul:

$$F(x_1, x_2, x_3, y) = R(x_1, x_2, x_3) + y(5x_1 + 7x_2 + 10x_3 - 1\,300)$$

Aici, am considerat relaţia I egalitate. Această deformare a problemei, se explică prin insuficienta eficienţă a metodei lagrangeanului, pentru tipurile de programe cu restricţii date prin inegalităţi.

Vom căuta, ulterior, cu preţul unei analize destul de laborioasă şi fără putere de acţiune suficient de largă, să înlăturăm această impunere artificială.

Scanned with OKEN Scanner

Obţinem în acest caz sistemul

$$5y + 10 - 28\,\Phi\left(\frac{x_1 - 80}{20}\right) = 0,$$

IV. $$7y + \frac{13}{2} - 19\,\Phi\left(\frac{x_2 - 40}{12}\right) = 0,$$

$$10y + 12 - 34\,\Phi\left(\frac{x_3 - 50}{15}\right) = 0,$$

$$5x_1 + 7x_2 + 10x_3 = 1\,300.$$

O analiză sumară a formei acestui sistem şi concluziile trase în urma rezolvării problemei fără restricţii, duc la concluzia că y trebuie să aibă o valoare negativă. Aceasta, deoarece soluţiile determinate anterior duceau la o depăşire a valorii 1 300 pentru expresia $(5x_1 + 7x_2 + 10x_3)$.

Or, pentru a remedia această defecţiune, trebuie ca soluţiile sistemului IV să fie mai mici decît cele ale sistemului III, şi aceasta se realizează doar pentru $y < 0$.

O modalitate de a determina soluţia sistemului IV ar fi următoarea: se pleacă de la un $y < 0$ arbitrar şi se rezolvă sistemul. Dacă $9x_1^* + 7x_2^* + 10x_3^* = 1\,300$, atunci s-a realizat soluţia optimă; dacă $9x_1^* + 7x_2^* + 10x_3^* > 1\,300$ mărim pe y, iar dacă $9x_1^* + 7x_2^* + 10x_3^* < 1\,300$ îl micşorăm.

Soluţia optimă nu va putea fi obţinută, însă, decît prin interpolare.

Pentru $y = -0{,}7$

$$x_1 = 94, \quad x_2 = 43, \quad x_3 = 56, \text{ ceace duce la}$$

$$5x_1 + 7x_2 + 10x_3 = 1\,331.$$

Pentru $y = -0{,}9$

$$x_1 = 91, \quad x_2 = 40, \quad x_3 = 53, \text{ şi deci}$$

$$5x_1 + 7x_2 + 10x_3 = 1265.$$

Putem lua, cu o bună aproximaţie, pentru soluţia optimă

$$x_1^* \simeq 93, \quad x_2^* \simeq 41, \quad x_3^* = 55.$$

Scanned with OKEN Scanner

Exemplul ilustrează, inconveniente destul de substanţiale ale metodei lagrangeanului, care o fac practic, inaplicabilă în marea majoritate a problemelor de programare neliniară.

Metode adecvate, cu eficienţă destul de mare, au fost elaborate, pînă în prezent, pentru anumite clase de program neliniare. În următoarele paragrafe ne vom referi la acestea.

§ 2. PROGRAMAREA CONVEXĂ

2.1. Considerente generale

O importantă clasă de programe neliniare o constituie aceea a programării convexe.

Acest domeniu al programării matematice are o dublă importanţă; atît datorită faptului că majoritatea problemelor de programare neliniară sugerate de modelele economice conţin funcţii convexe sau concave, cît şi de faptul că aici au fost elaborate metode matematice satisfăcătoare.

La o problemă de acest tip regăsim unele din proprietăţile matematice de bază, care fundamentează algoritmul simplex — această eficientă metodă de rezolvare a programelor liniare.

În plus, teoria elaborată de Kuhn şi Tucker reglementează problema existenţei soluţiilor optime.

În scopul de a prezenta punctele esenţiale ale acestei teorii, să definim programul convex după cum urmează:

(1) Să se minimizeze funcţia convexă $f(X)$ în condiţiile:

(2) $$g_i(X) \leqslant 0, \qquad i = 1, 2, \ldots, m;$$

(3) $$x_j \geqslant 0, \qquad j = 1, 2, \ldots, n,$$

$g_i(X)$, $(i = 1, 2, \ldots, m)$, fiind funcţii convexe de n variabile reale $x_1, x_2, \ldots, x_n$.

Programul poate fi, de asemenea, scris sub forma concentrată

(4)

$$\min \{f(X) \mid x_j \geqslant 0,\ j = 1, 2, \ldots, n;\ g_i(X) \leqslant 0,\ i = 1, 2, \ldots, m\}$$

Scanned with OKEN Scanner

Putem remarca că sistemul de restricţii (2) nu înseamnă o restrîngere a generalităţii.

O restricţie de forma

$$g_i(X) \geqslant 0$$

poate fi adusă la forma din (2), punînd

$$\bar{g}_i(X) = -g_i(X) \text{ şi obţinînd deci}$$

$$\bar{g}_i(X) \leqslant 0.$$

De asemenea o egalitate de forma

$$g_i(X) = 0$$

este echivalentă cu sistemul de două inegalităţi

$$g_i(X) \leqslant 0,$$

$$g_i(X) \geqslant 0.$$

Trebuie menţionat faptul că specificul unui program convex constă în minimizarea unei funcţiei convexe, sau în maximizarea unei funcţii concave, cu condiţia ca mulţimea soluţiilor posibile X, să fie un domeniu convex.

Ca atare, alte formulări posibile ale unei probleme de programare convexă, ce pot fi tratate în aceeaşi manieră ca cea iniţial definită, ar fi

(4′) $\max\{f(X) | x_j \geqslant 0, \quad j = 1, 2, \ldots, n; \quad g_i(X) \geqslant 0, \quad i = 1, 2, \ldots, m\}$

unde $f(X)$ şi $g_i(X)$ $(i = 1, 2, \ldots, m)$ sînt funcţii concave de n variabile reale $x_1, x_2, \ldots, x$.

(4″) $\min\{fX) | x_j \geqslant 0, \quad j = 1, 2, \ldots, n; \quad g_i(X) \geqslant 0, \quad i = 1, 2, \ldots, m\}$ unde $f(X)$ este funcţia convexă, iar $g_i(X)$ $(i = 1, 2, \ldots, m)$ sînt funcţii concave.

(4) $\max\{f(X) | x_j \geqslant 0, \quad j = 1, 2, \ldots, n; \; g_i(X) \leqslant 0, \quad i = 1, 2, \ldots, n\}$

unde $f(X)$ este funcţie concavă iar $g_i(X)$, $(i = 1, 2, \ldots, m)$ sînt funcţii convexe.

Scanned with OKEN Scanner

Toate aceste programe pot fi cu uşurinţă aduse la forma (4) făcînd observaţia că dacă $\varphi(X)$ este o funcţie concavă, atunci $-\varphi(X)$ va fi convexă.

Problemele de maximizare a unei funcţii concave, respectiv, de minimizare a unei funcţii concave, nu se încadrează în clasa programării convexe.

Dăm în continuare cîteva definiţii:

Fie $\mathfrak{X}$ mulţimea soluţiilor posibile pentru programul (4)

$$(\mathfrak{X} = \{X | x_j \geqslant 0, \quad j = 1, 2, \ldots, n; \; g_i(X) \leqslant 0, \quad i = 1, 2, \ldots, m\})$$

Programul se numeşte *admisibil* dacă $\mathfrak{X} \neq \Phi$.

Mulţimile de puncte $\mathfrak{X}$, satisfăcînd o condiţie de tipul $g_i(X) = 0$, sau de tipul $x_j = 0$, se vor numi *hiperplane* sau *hipersuprafeţe — frontieră*, ale domeniului $\mathfrak{X}$.

Un punct admisibil $X \in \mathfrak{X}$ se va numi *punct frontieră* dacă se va găsi pe cel puţin o hipersuprafaţă — frontieră a lui $\mathfrak{X}$.

Un punct admisibil $X \in \mathfrak{X}$ se va numi *punct interior* al mulţimii soluţiilor posibile $\mathfrak{X}$, dacă nu aparţine nici unei hipersuprafeţe — frontieră.

Este clar că pentru un astfel de punct

$$x_j > 0, \qquad j = 1, 2, \ldots, n;$$
$$g_i(X) < 0, \qquad i = 1, 2, \ldots, m.$$

Dăm în continuare cîteva rezultate imediate, care precizează unele proprietăţi ale soluţiilor problemei.

Teorema 1. *Mulţimea $\mathfrak{X}$ a soluţiilor posibile ale programului (4) este convexă.*

Într-adevăr dacă $\mathfrak{X} \neq \Phi$, fie $X^1, X^2 \in \mathfrak{X}$ şi $\lambda \in [0, 1]$.

Din definiţia convexităţii funcţiei g_i, rezultă

$$g_i[\lambda X^1 + (1-\lambda)X^2] \leqslant \lambda g_i(X^1) + (1-\lambda) g_i(X^2) \leqslant 0, \quad i = 1, 2, \ldots, m.$$

deoarece X^1 şi X^2 satisfac (2).

De asemenea, evident

$$\lambda x_j^1 + (1 - \lambda) x_j^2 \geqslant 0, \qquad j = 1, 2, \ldots, n$$

ceea ce probează faptul că $\lambda X^1 + (1 - \lambda) X^2 \in \mathfrak{X}$.

Scanned with OKEN Scanner

Teorema 2. *Mulțimea soluțiilor optime ale programului convex* (4) *este convexă. Dacă f este strict convexă, atunci soluția optimă* (*dacă există*) *este unică.*

Presupunem că programul considerat are soluții optime. Fie X^* și X^{**} două dintre acestea și $\lambda \in [0, 1]$.

Din definiția optimalității:

$$f(X^*) \leqslant f(X) \text{ și } f(X^{**}) \leqslant f(X), \text{ pentru orice } X \in \mathcal{X}.$$

Combinînd aceste inegalități cu definiția convexității lui f,

$f[\lambda X^* + (1 - \lambda)X^{**}] \leqslant \lambda f(X^*) + (1 - \lambda)f(X^{**}) \leqslant f(X)$, pentru orice $X \in \mathcal{X}$.

Adică, $\lambda X^* + (1 - \lambda)X^{**}$ este soluție optimă.

Dacă f este strict convexă, relația de mai sus devine $f[\lambda X^* + (1 - \lambda)X^{**}] < f(X)$, pentru orice $X \in \mathcal{X}$, și în particular

$$f[\lambda X^* + (1 - \lambda)X^{**}] < f(X^*),$$

ceea ce ar contrazice optimalitatea lui X^*.

Fie acum X^* o soluție optimă a programului (4).

Să introducem notațiile

$$\hat{N} = \{j | 1 \leqslant j \leqslant n,\ x_j^* = 0\}$$

$$\hat{M} = \{i | 1 \leqslant i \leqslant m,\ g_i(X^*) = 0\}.$$

Bine înțeles, dacă $j \notin \hat{N}$, $x_j^* > 0$ și dacă $i \notin \hat{M}$, $g_i(X^*) < 0$.

Teorema 3. *Dacă X^* este soluție optimă a problemei* (4) *atunci este o soluție optimă a programului restrîns*

$$\min \{f(X) |\ x_j \geqslant 0,\ j \in \hat{N};\ g_i(X) \leqslant 0,\ i \in \hat{M}|. \tag{5}$$

Presupunînd că X^* nu ar verifica (5), atunci ar exista $\overline{X}$

$$\bar{x}_j \geqslant 0 \text{ pentru } j \in \hat{N},$$

$$g_i(\overline{X}) \geqslant 0 \text{ pentru } i \in \hat{M},$$

$$f(\overline{X}) < f(X^*).$$

Scanned with OKEN Scanner

Fie $0 < \lambda < 1$. Atunci

$$g_i[\lambda\bar{X} + (1 - \lambda)X^*] \leqslant \lambda g_i(\bar{X}) + (1 - \lambda)g_i(X^*) \leqslant 0,$$
$$i = 1, 2, \ldots, m.$$

Într-adevăr, pentru $i \in \hat{M}$, inegalitatea este banală deoarece atît $g_i(\bar{X}) \leqslant 0$ cît și $g_i(X^*) \leqslant 0$.

Pentru $i \notin \hat{M}$ inegalitatea poate fi încă verificată cu condiția ca λ să fie suficient de mic, astfel încît termenul $\lambda g_i(\bar{X})$ să nu influențeze semnul sumei ce apare pe locul central.

Într-o manieră asemănătoare se verifică

$$\lambda\bar{x}_j + (1 - \lambda)x_j^* \geqslant 0, \quad j = 1, 2, \ldots, n.$$

Obținem deci, pentru o valoare suficient de mică a lui λ, o soluție posibilă $\lambda\bar{X} + (1 - \lambda)X^*$ a problemei (4).

În plus

$$f[\lambda\bar{X} + (1 - \lambda)X^*] \leqslant \lambda f(\bar{X}) + (1 - \lambda)f(X^*) < f(X^*)$$

Or, această ultimă inegalitate contrazice optimalitatea lui X^* pentru programul (4).

Dăm, fără demonstrație și următoarea:

Teorema 4. *Fie X^* o soluție optimă a problemei* (4). *Atunci mulțimea soluțiilor optime este formată din toate punctele $\bar{X} \in \mathcal{X}$ verificînd condițiile*

$$\frac{\partial f}{\partial x_j}(\bar{X}) = \frac{\partial f}{\partial x_j}(X^*), \qquad j = 1, 2, \ldots, n; \tag{6}$$

$$\sum_{j=1}^{n} (\bar{x}_j - x_j^*) \frac{\partial f}{\partial x_j}(X^*) = 0. \tag{7}$$

2.2. Condițiile Kuhn-Tucker

Teorema Kuhn-Tucker ocupă un loc central în teoria programării convexe.

Generalizînd metoda lagrangeanului la programe în care restricțiile sînt date sub formă de inegalități, ea permite formu-

Scanned with OKEN Scanner

larea unor condiţii necesare şi suficiente pentru existenţa soluţiilor optime.

Importanţa acestei teoreme este, atît de natură teoretică, cît şi practică, majoritatea algoritmilor de rezolvare a diferitelor tipuri de probleme de programare convexă, avînd aici punctul de plecare.

Vom prezenta, în continuare, acest rezultat fundamental într-o formă modificată prin înlocuirea condiţiilor de regularitate originare prin altele datorate lui Slater.

Fie

$$F(X, Y) = f(X) + \sum_{i=1}^{m} y_i g_i(X).$$

Teorema 5. *Vectorul* $X^* = (x_1^*, x_2^*, \ldots, x_n^*)$ *este o soluţie optimă a problemei* (4) *dacă şi numai dacă există un vector* $Y^* = (y_1^*, y_2^*, \ldots, y_m^*)$ *astfel ca*

(8) $$x_j^* \geqslant 0, \qquad j = 1, 2, \ldots, n;$$

(9) $$y_i^* \geqslant 0, \qquad i = 1, 2, \ldots, m;$$

(10) $$F(X^*, Y) \leqslant F(X^*, Y^*) \leqslant F(X, Y^*)$$

pentru orice X *şi* Y *cu* $x_j \geqslant 0$, $j = 1, 2, \ldots, n$; $y_i \geqslant 0$, $i = 1, 2, \ldots, m$;

(11) *există* $\overline{X} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ *astfel că*

$$\bar{x}_j \geqslant 0, \qquad j = 1, 2, \ldots, n,$$

$$g_i(\overline{X}) < 0, \qquad i = 1, 2, \ldots, m.$$

Deoarece demonstraţia completă a teoremei depăşeşte cadrul acestei lucrări, ne vom limita la demonstrarea suficienţei condiţiilor.

Fie deci X^*, Y^* satisfăcînd (8), (9) şi (10).

Ultima condiţie se transformă în

(12) $$f(X^*) + \sum_{i=1}^{m} y_i g_i(X^*) \leqslant f(X^*) + \sum_{i=1}^{m} y_i^* g_i(X^*) \leqslant f(X) + \sum_{i=1}^{m} y_i^* g_i(X).$$

Deoarece inegalitatea din stînga este verificată pentru orice Y cu $y_i \geqslant 0$, putem alege, pentru fiecare $i = 1, 2, \ldots, m$, un $\overline{Y}$ cu $\overline{y_k} = y_k^* + \delta_{ki}\alpha$, $(\alpha > 0)$ şi atunci, obţinem

$$g_i(X^*) \leqslant 0, \qquad i = 1, 2, \ldots, n.$$

Scanned with OKEN Scanner

Aceasta, împreună cu (8) implică $X^* \in \mathcal{X}$ şi, totodată

$$\sum_{i=1}^{m} y_i^* g_i(X^*) \leqslant 0.$$

Alegem acum $Y = 0$ în partea stîngă a relaţiei (12) şi obţinem

$$0 \leqslant \sum_{i=1}^{m} y_i^* g_i(X^*).$$

Deci

$$\sum_{j=1}^{m} y_i^* g_i(X^*) = 0.$$

Atunci inegalitatea din dreapta din (12) se transformă în

$$f(X^*) \leqslant f(X) + \sum_{i=1}^{m} y_i^* g_i(X).$$

Dacă $X \in \mathcal{X}$, atunci $g_i(X) \leqslant 0$, $(i = 1, 2, \ldots, m)$, şi cum $y_i^* \geqslant 0$, $(i = 1, 2, \ldots, m)$ rezultă

$$f(X^*) \leqslant f(X)$$

ceea ce nu înseamnă altceva decît că X^* este soluţie optimă a problemei.

Condiţiile (8)—(10) ale teoremei stipulează, de fapt, existenţa unui punct şa nenegativ pentru funcţia lagrangean $F(X, Y)$.

În ipoteza suplimentară $F \in C^1$, se poate arăta că aceste condiţii sînt echivalente cu următoarele condiţii, introduse iniţial de Kuhn şi Tucker:

$$\frac{\partial F}{\partial x_j}(X^*, Y^*) \geqslant 0, \qquad j = 1, 2, \ldots, n,$$

$$(13) \qquad x_j^* \frac{\partial F}{\partial x_j}(X^*, Y^*) = 0, \qquad j = 1, 2, \ldots, n,$$

$$x_j^* \geqslant 0, \qquad j = 1, 2, \ldots, n,$$

$$\frac{\partial F}{\partial y_i}(X^*, Y^*) \leqslant 0, \qquad i = 1, 2, \ldots, m,$$

$$(14) \qquad y_i^* \frac{\partial F}{\partial y_i}(X^*, Y^*) = 0, \qquad i = 1, 2, \ldots, m,$$

$$y_i^* \geqslant 0 \qquad i = 1, 2, \ldots, m.$$

Scanned with OKEN Scanner

Într-adevăr, dacă (8)—(10) sînt satisfăcute, atunci în baza celor deduse în § 1 (acest capitol), rezultă şi valabilitatea condiţiilor (13), (14) (vezi relaţiile (19′), (20′) din § 1).

Reciproc, dacă (13) şi (14) sînt verificate, atunci în virtutea convexităţii în X a lui f şi implicit a lui F,

$$F(X, Y^*) \geqslant F(X^*, Y^*) + \sum_{j=1}^{n} (x_j - x_j^*) \frac{\partial F}{\partial x_j}(X^*, Y^*) = \tag{15}$$
$$= F(X^*, Y^*) + \sum_{j=1}^{n} x_j \frac{\partial F}{\partial x_j}(X^*, Y^*).$$

Dacă X are componentele nenegative, atunci, datorită lui (13), (15) se poate scrie

$$F(X, Y^*) \geqslant F(X^*, Y^*)$$

Cum F este liniară în Y, avem

$$F(X^*, Y) = F(X^*, Y^*) + \sum_{i=1}^{m} (y_i - y_i^*) \frac{\partial F}{\partial y_i}(X^*, Y^*) = \tag{16}$$
$$= F(X^*, Y^*) + \sum_{i=1}^{m} y_i \frac{\partial F}{\partial y_i}(X^*, Y^*).$$

Dacă Y are componentele nenegative, atunci (16) combinat cu (4) dă

$$F(X^*, Y) \leqslant F(X^*, Y^*).$$

În concluzie, am demonstrat că

$$F(X^*, Y) \leqslant F(X^*, Y^*) \leqslant F(X, Y^*)$$

pentru X^*, Y^* cu $x_j^* \geqslant 0$ şi $y_i^* \geqslant 0$ şi pentru toţi X cu $x_j \geqslant 0$ şi Y cu $y_i \geqslant 0$, adică F are un punct şa nenegativ.

Atragem atenţia asupra faptului că în demonstraţia dată pentru suficienţa condiţiilor teoremei 5 nu s-a folosit condiţia (11).

Această ipoteză de regularitate, care înlocuieşte pe cea introdusă iniţial de Kuhn şi Tucker, joacă însă un rol esenţial în demonstrarea necesităţii condiţiilor.

Scanned with OKEN Scanner

2.3. Rezolvarea programelor convexe: metoda gradientului proiectat a lui Rosen

Marele avantaj prezentat de problemele programării convexe este dat de faptul că pentru o funcţie convexă (concavă), orice minim (maxim) local este şi global. Dacă în plus convexitatea (concavitatea) este strictă, optimul este unic.

Acest aspect a permis elaborarea unor algoritme, bazate pe teoria Kuhn-Tucker, care duc la soluţia optimă.

De menţionat, că folosirea algoritmelor respective în probleme generale de programare neliniară poate duce la obţinerea unor soluţii locale care să nu fie, totodată, soluţii optime globale.

Să urmărim în continuare, prezentarea succintă a unei astfel de metode de rezolvare a problemelor de programare convexă, elaborată pentru un tip mai restrîns de programe de acest gen.

Vom aborda problema sub forma de maximizare:

$$(17) \qquad \max\{f(X)/g_i(X) \leqslant 0, \quad i = 1, 2, \ldots, m\}$$

unde

$$(18) \quad g_i(X) = \sum_{k=1}^{n} a_{ik}x_k - b_i, \; i = 1, 2, \ldots, m, \; a_{ik}, \; b_i \in R,$$

iar f este o funcţie concavă.

Dată fiind forma liniară a restricţiilor, putem presupune că şi condiţiile de nenegativitate a variabililor sînt incluse în sistemul

$$(19) \qquad g_i(X) \leqslant 0, \qquad i = 1, 2, \ldots, m.$$

Înainte de a descrie algoritmul, vom face o scurtă prezentare a criteriului de optimalitate care stă la baza lui (care derivă, de fapt, direct din condiţiile Kuhn-Tucker).

Cu ocazia aceasta, în scopul scurtării expunerii, vom introduce unele noţiuni noi şi vom reformula, în alţi termeni noţiuni anterior definite, încadrîndu-le într-un, aşa zis, limbaj geometric.

O hipersuprafaţă — frontieră, în cazul problemei considerată, devine un hiperplan — frontieră, de tipul

$$H_i = \left\{X \left| \sum_{k=1}^{n} a_{ik}x_k - b_i = 0 \right.\right\} = \{X \,|a_i\, X - b_i = 0\}$$

unde am notat

Scanned with OKEN Scanner

$a_i = (a_{i1}, a_{i2}, \ldots, a_{in})$ (X este considerat vector coloană). Vectorul $a_i = (a_{i1}, \ldots, a_{in})$ poate fi interpretat „geometric" ca vectorul normal pe hiperplanul H_i orientat spre exteriorul domeniului punctelor admisibile $\mathfrak{X}$.

Notăm cu A matricea

$$\left\| \begin{matrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{matrix} \right\|$$

şi presupunem că rangul lui A este r.

Fără a restrînge generalitatea, putem presupune că primii r vectori linie ai matricei A, $a_1, a_2, \ldots, a_r$ sînt liniar independenţi

Fie $D = \left\{ X ; \sum_{k=1}^{n} a_{ik} x_k - b_i = 0, \quad i = 1, 2, \ldots, r \right\} = \bigcap_{i=1}^{r} H_i$.

Atunci D, ca intersecţie a celor r hiperplane liniar independente, este un subspaţiu liniar, $(n - r)$-dimensional, al lui R^n.

Fie $\tilde{D}$ subspaţiul vectorial generat de vectorii liniar independenţi $a_1, a_2, \ldots, a_r$, adică

$$\tilde{D} = \left\{ X \in R^n | X = \sum_{i=1}^{r} y_i a_i \right\}.$$

Cum, fiecare a_i este perpendicular pe hiperplanul $H_i (i = 1, 2, \ldots, r)$, rezultă că D şi $\tilde{D}$ sînt subspaţii liniare ortogonale.

De aici rezultă că fiecare vector $X \in R^n$ se poate reprezenta sub forma

$$X = X_D + X_{\tilde{D}} \text{ cu } X_D \in D \text{ şi } X_{\tilde{D}} \in \tilde{D}.$$

X_D şi $X_{\tilde{D}}$ se vor numi proiecţiile vectorului X pe subspaţiile liniare D, respectiv $\tilde{D}$.

Scanned with OKEN Scanner

Cele două proiecții X_D și $X_{\tilde{D}}$ se pot obține, înmulțind la stînga vectorul X cu matricele de proiecție

$$P_r = E - A_r' (A_r A_r')^{-1} A_r,$$

$$P_r = A_r' (A_r A_r')^{-1} A_r$$

unde $A_r = \left\| \begin{matrix} a_1 \\ a_2 \\ \vdots \\ a_r \end{matrix} \right\|$ și E este matricea unitate n-dimensională.

Dacă pentru un $X \in R^n$, $P_r X = X_D = 0$, rezultă din reprezentarea $X = X_D + X_{\tilde{D}}$, că $X \in \tilde{D}$.

Fie acum $X^* \in D$. Folosind condițiile Kuhn-Tucker (13)—(14) ne propunem să deducem o expresie simplă a condițiilor necesare și suficiente ca X^* să fie o soluție optimă a problemei (17).

Pentru aceasta va fi necesar să aducem problema (17) la forma (4), pentru care au fost deduse (13)—(14).

Acest deziderat poate fi realizat înlocuind condiția de maxim asupra lui f prin condiția de minim asupra lui $(-f)$(max $f =$ $= -$ min $(-f)$).

Atunci

$$F(X, Y) = -f(X) + \sum_{i=1}^{m} y_i \left[\sum_{k=1}^{m} a_{ik} x_k - b_i \right],$$

$$\frac{\partial F}{\partial x_j}(X, Y) = -\frac{\partial f}{\partial x_j}(X) + \sum_{i=1}^{m} y_i a_{ij},$$

$$\frac{\partial F}{\partial y_i}(X, Y) = \sum_{k=1}^{n} a_{ik} \cdot x_k - b_i.$$

Să începem prin a transcrie condițiile (14):

$$\sum_{k=1}^{n} a_{ik} x_k^* - b_i \leqslant 0, \qquad i = 1, 2, \ldots, m;$$

$$(20) \qquad y_i^* \left[\sum_{k=1}^{n} a_{ik} x_k - b_i \right] = 0, \qquad i = 1, 2, \ldots, m;$$

$$y_i^* \geqslant 0, \qquad i = 1, 2, \ldots, m.$$

Deoarece am presupus $X^* \in D$, (20) devine

$$\sum_{k=1}^{n} a_{ik}x_k^* - b_i = 0, \qquad i = 1, 2, \ldots, r;$$

$$\sum_{k=1}^{n} a_{ik}x_k^* - b_i < 0, \qquad i = r+1, \ldots, m; \tag{21}$$

$$y_i^* \geqslant 0, \qquad i = 1, 2, \ldots, r;$$

$$y_i^* = 0, \qquad i = r+1, \ldots, m.$$

În fond, (21) nu spune altceva decît că X^* este un punct admisibil şi că primii r multiplicatori ai lui Lagrange sînt nenegativi, iar cealalţi nuli.

Ţinînd seama de faptul, că în (17), restricţiile de nenegativitate, au fost incluse în sistemul restricţiilor (19), condiţiile (13) devin

$$-\frac{\partial f}{\partial x_j}(X^*) + \sum_{i=1}^{r} y_i^* a_{ij} = 0, \qquad j = 1, 2, \ldots, n. \tag{22}$$

De aici şi din (21) avem

$$\frac{\partial f}{\partial x_j}(X^*) = \sum_{i=1}^{r} y_i^* a_{ij}, \qquad j = 1, 2, \ldots, n. \tag{23}$$

Folosind clasica notaţie a gradientului unei funcţii

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right)$$

relaţiile (23) şi (21), exprimă faptul că vectorul

$\nabla f(X^*)$, este o combinaţie liniară, cu coeficienţi nenegativi $y_1^*, y_2^*, \ldots, y_r^*$, de vectorii $a_1, a_2, \ldots, a_r$, adică $\nabla f(X^*) \in \tilde{D}$, ceea ce se poate scrie concentrat

$$P_r \nabla f(X^*) = 0,$$

$$(A_r A_r')^{-1} A_r \nabla f(X^*) \geqslant 0. \tag{24}$$

Putem atunci enunţa un criteriu de optimalitate pentru problema (17):

Condiția necesară și suficientă, pentru ca un punct admisibil frontieră X^*, aflat la intersecția a r hiperplane liniar independente, să fie soluție optimă, este ca relațiile (24) *să fie satisfăcute.*

Iată acum, etapele algoritmului:

În prima etapă se determină un punct X° admisibil, situat la intersecția a r hiperplane-frontieră (r fiind rangul matricei A).

Aceasta se poate obține, rezolvînd un sistem de r ecuații independente, obținute prin transformarea inegalităților din (19) în egalități (sau eventual folosind algoritmul simplex, pentru determinarea unui punct extrem).

Se calculează $\nabla f(X^0)$ și P_r.

Dacă (24) sînt satisfăcute, atunci X^0 este soluție optimă.

În caz contrar se caută o îmbunătățire a soluției, înlocuindu-l pe X^0 cu X^1.

În general, dacă după k etape X^k nu este soluție optimă, construcția soluției îmbunătățite X^{k+1} se face astfel:

a) Dacă $P_r \nabla f(X^k) \neq 0$, punem

$$s = P_r \nabla f(X^k).$$

b) Dacă $P_r \nabla f(X^k) = 0$, dar $u = (A_r A_r')^{-1} A_r \nabla f(X^k)$, are cel puțin o componentă $u_t < 0$, punem

$$s = P_{r-1} \nabla f(X^k),$$

P_{r-1} fiind matricea de proiecție corespunzătoare subspațiului liniar $D' = \bigcap_{\substack{i=1 \\ i \neq t}}^{r} H_i$.

În continuare, cu s determinat, se alege

$$\lambda = \min_{\lambda_j > 0} \left\{ \lambda_j | \lambda_j = -\frac{g_j(X^k)}{a_j s}, \quad j = r+1, \ldots, m \right\}$$

și se scrie

$$\overline{X}^k = X^k + \lambda s.$$

Dacă $s' \nabla f(\overline{X}^k) \geq 0$, se alege

$$X^{k+1} = \overline{X}^k.$$

Scanned with OKEN Scanner

Dacă $s'\nabla f(X^k) < 0$, se alege

$$X^{k+1} = \rho \bar{X}^k + (1 - \rho) X^k$$

$$\text{cu } \rho = \frac{s'\nabla f(X^k)}{s'\nabla f(X^k) - s'\nabla f(\bar{X}^k)}.$$

Dacă X^{k+1}, astfel determinat, nu este soluție optimă, se repetă procedeul.

Vom ilustra metoda descrisă printr-un exemplu numeric.

Fie programul convex:

Să se maximizeze

$$f(X) = -\frac{1}{2}x_1^2 - \frac{1}{2}x_2^2 + x_1 + 2x_2,$$

variabilele x_1, x_2 *satisfăcînd restricțiile*

$$\begin{aligned} 2x_1 + 3x_2 &\leqslant 6, \\ x_1 + 4x_2 &\leqslant 5, \\ x_1 &\geqslant 0, \\ x_2 &\geqslant 0. \end{aligned}$$

Verificarea concavității lui f este imediată $\left(\frac{\partial^2 f}{\partial x_1^2} < 0, \frac{\partial^2 f}{\partial x_2^2} < 0\right)$.

Cu notațiile introduse anterior;

$a_1 = (2, 3)$, $a_2 = (1,4)$, $a_3 = (-1, 0)$ $a_4 = (0, -1)$,

$$A = \begin{Vmatrix} 2 & 3 \\ 1 & 4 \\ -1 & 0 \\ 0 & -1 \end{Vmatrix}, \text{ rang } (A) = 2, \quad r = 2,$$

$$\nabla f(X) = \begin{Vmatrix} -x_1 + 1 \\ -x_2 + 2 \end{Vmatrix}.$$

Prima etapă

Punem $A_2 = \begin{Vmatrix} 2 & 3 \\ 1 & 4 \end{Vmatrix}$

$$A_2^{-1} = \begin{Vmatrix} 4/5 & -3/5 \\ -1/5 & 2/5 \end{Vmatrix};$$

$$P_2 = E - A_2'(A_2 A_2')^{-1} \cdot A_2 = E - A_2'(A_2')^{-1} A_2^{-1} A_2 = 0$$

(acest lucru se datorește formei pătrate particulare, a lui A_2).

Scanned with OKEN Scanner

Calculăm pe X° din sistemul

$$2x_1 + 3x_2 = 6,$$
$$x_1 + 4x_2 = 5.$$

Rezultă

$$X^0 = \begin{pmatrix} 9/5 \\ 4/5 \end{pmatrix}.$$

Evident $P_2 - \nabla f(X^0) = 0$,

$$U = (A_2 A_2')^{-1} A_2 \nabla f(X^0) = (A_2')^{-1} \nabla f(X^0) =$$
$$= \begin{Vmatrix} 4/5 & -1/5 \\ -3/5 & 2/5 \end{Vmatrix} \begin{pmatrix} -4/5 \\ 6/5 \end{pmatrix} = \begin{pmatrix} -22/25 \\ 24/25 \end{pmatrix}.$$

Deoarece $u_1 = -\frac{22}{25} < 0$, X^0 nu este optim.

A doua etapă

În conformitate cu b) se elimină prima linie din A_2. Obţinem:

$$A_1 = (1, 4),$$
$$A_1 A_1' = 17, \qquad (A_1 A_1')^{-1} = \frac{1}{17},$$
$$A_1'(A_1 A_1')^{-1} A_1 = \begin{Vmatrix} 1/17 & 4/17 \\ 4/17 & 16/17 \end{Vmatrix},$$
$$P_1 = E - A_1'(A_1 A_1') A_1 = \begin{Vmatrix} 16/17 & -4/17 \\ 4/17 & 1/17 \end{Vmatrix}$$

Deci

$$s = P_1 \nabla f(X^0) = \begin{pmatrix} -88/85 \\ 22/85 \end{pmatrix};$$

$$\lambda_1 = -\frac{g_1(X^0)}{a_1 s} = 0, \qquad \lambda_2 = -\frac{g_2(X^0)}{a_2 s} = 0,$$
$$\lambda_3 = -\frac{g_3(X_0)}{a_3 s} = -\frac{-9/5}{88/85} = \frac{153}{88},$$
$$\lambda_4 = -\frac{g_4(X_0)}{a_4 s} = -\frac{-4/5}{22/85} = -\frac{34}{11}.$$

Scanned with OKEN Scanner

Ca atare

$$\lambda = \lambda_3 = \frac{153}{88}.$$

Atunci

$$\overline{X}^0 = \begin{pmatrix} 9/5 \\ 4/5 \end{pmatrix} + \frac{153}{88}\begin{pmatrix} -88/85 \\ 22/85 \end{pmatrix} = \begin{pmatrix} 0 \\ 5/4 \end{pmatrix},$$

$$\nabla f(\overline{X}^0) = \begin{pmatrix} 1 \\ 3/4 \end{pmatrix}.$$

În consecință

$$s' \nabla f(\overline{X}^0) = \left(-\frac{88}{85}, \frac{22}{85}\right)\begin{pmatrix} 1 \\ 3/4 \end{pmatrix} = -\frac{143}{170} < 0.$$

Vom calcula atunci

$$\rho = \frac{\left(-\frac{88}{85}, \frac{22}{85}\right)\cdot\begin{pmatrix} -4/5 \\ 6/5 \end{pmatrix}}{\left(-\frac{88}{85}, \frac{22}{85}\right)\cdot\begin{pmatrix} -4/5 \\ 6/5 \end{pmatrix} + \frac{143}{170}} = \frac{88}{153}.$$

Atunci, noul vector obținut va fi

$$X^1 = \frac{88}{153}\begin{pmatrix} 0 \\ 5/4 \end{pmatrix} + \frac{65}{153}\begin{pmatrix} 9/5 \\ 4/5 \end{pmatrix} = \begin{pmatrix} 13/17 \\ 18/17 \end{pmatrix}.$$

Verificăm dacă X^1 este optim;

$$P_1 \nabla f(X^1) = \begin{Vmatrix} 16/17 & -4/17 \\ -4/17 & 1/17 \end{Vmatrix}\begin{pmatrix} 4/17 \\ 16/17 \end{pmatrix} = 0,$$

$$U = (A_1 A_1')^{-1} A_1 \nabla f(X^1) = 1/17(1,4)\begin{pmatrix} 4/17 \\ 16/17 \end{pmatrix} = \frac{68}{289} > 0.$$

Prin urmare, soluția optimă a problemei este

$$x_1^* = \frac{13}{17}, \quad x_2^* = \frac{18}{17}$$

Scanned with OKEN Scanner

§ 3. PROGRAMAREA PĂTRATICĂ

3.1. Forma generală a unui program pătratic. Exemplu

Programarea pătratică constituie acea clasă a problemelor de programare neliniară căreia i-au fost consacrate cele mai numeroase studii, răsplătite, unele, cu rezultate eficiente, comparabile din acest punct de vedere cu metodele de rezolvare a programelor liniare.

Printr-o problemă de programare pătratică se înţelege un program în care restricţiile sînt liniare, iar funcţia obiectiv este suma dintr-o funcţională liniară şi o formă pătratică.

Forma „standard" pe care o vom prezenta aici, este următoarea:

$$\min \{f(X) \mid AX \leqslant b,\ X \geqslant 0\}. \tag{1}$$

Principalele elemente conţinute în acest program sînt astfel definite:

$A = (a_{ij})_{\substack{1 \leqslant i \leqslant m \\ 1 \leqslant j \leqslant n}}$ este o matrice de ordinul $(m \times n)$;

b este un vector cu componentele b_i $(i = 1, 2, \ldots, m)$;

$$f(X) = dX + X' CX$$

unde $d = (d_1, d_2, \ldots, d_n)$, iar

$C = (c_{ij})_{\substack{1 \leqslant i \leqslant n \\ 1 \leqslant j \leqslant n}}$ este o matrice, pătrată, simetrică, nenegativ definită.

Cîteva observaţii se impun.

Proprietăţile lui C, fac ca $f(X)$ să fie o funcţie convexă şi plasează programul pătratic în clasa programelor convexe, ca un caz particular.

Atunci, toate rezultatele şi discuţiile din § 2 sînt valabile în cazul problemei (1), însă cu unele îmbunătăţiri datorate caracterului particular al acesteia.

Dacă C este pozitiv definită, atunci $f(X)$ va fi strict convexă, şi de aici, va rezulta şi unicitatea soluţiei optime.

Ipoteza de simetrie impusă matricii C nu înseamnă o restrîngere a generalităţii.

Scanned with OKEN Scanner

Într-adevăr, pentru forma pătratică

$$X'AX = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j$$

se poate verifica imediat egalitatea

$$X'AX = X'BX,$$

unde

$$B = (b_{ij})_{\substack{1 \leqslant i \leqslant n \\ 1 \leqslant j \leqslant n}} \quad \text{cu} \quad b_{ij} = b_{ji} = \frac{a_{ij} + a_{ji}}{2}.$$

Forma particulară (liniară) a restricțiilor problemei de programare pătratică face posibilă, folosind tehnica transformărilor echivalente a sistemelor de ecuații și inecuații liniare, reformarea problemei 1) sub aspectele

1') $\min \{f(X) | AX = b, \quad X \geqslant 0\}$,

1'') $\min \{f(X) | AX \geqslant b, \quad X \geqslant 0\}$,

1''') $\min \{f(X) | AX \leqslant b\}$.

În ultima expresie, condițiile de nenegativitate a variabilelor (dacă se pun) se consideră înglobate în sistemul de inecuații $AX \leqslant b$.

În sfîrșit putem nota că, dacă matricea C este nepozitiv definită, cum acest fapt implică concavitatea funcției — obiectiv f, problema pătratică poate fi formulată și cu cerințe de maxim, așa cum rezultă din discuția cazului general al programului convex (§ 2).

Un exemplu simplu de program pătratic, care se pretează la reprezentarea grafică clară, am dat în § 1 (ex. II).

Probleme care conduc la programări pătratice, apar deseori în economie sau știință.

Vom cita pentru exemplificare un model economic de largă circulație, acela al investițiilor.

Să presupunem că se urmărește un plan de investiții optime al unei sume globale S în n activități productive.

Evident, aprecierea sumelor $x_i (i = 1, 2, \ldots, n)$ ce vor fi afectate activităților productive i, trebuie să se facă de așa manieră încît să asigure un beneficiu maxim.

Scanned with OKEN Scanner

De cele mai multe ori beneficiul unitar r_i, dat de activitatea i, nu este constant, ci are o comportare aleatoare. Mai mult, practică economică relevă faptul că între beneficiile corespunzătoare mai multor ramuri productive există o interdependenţă.

Să presupunem că prin metode statistice, s-au determinat valorile medii m_i ale beneficiilor unitare în activităţile productive i ($i = 1, 2, \ldots, n$), şi corelaţiile σ_{ij} existente între beneficiile corespunzătoare a două ramuri productive i şi j ($i, j = 1, 2, \ldots, n$). În asemenea condiţii investitorul este obligat să ia în consideraţie valoarea medie a beneficiului total realizat în urma politicii sale de investiţii $X = (x_1, x_2, \ldots, x_n)$.

Aceasta va fi

$$M = \sum_{i=1}^{n} m_i x_i.$$

O politică optimă trebuie, evident, să maximizeze pe M.

În acelaşi timp, ţinînd seama de eventualele fluctuaţii (date de esenţa aleatoare a fenomenului), investitorul doreşte să obţină o stabilitate maximă a beneficiului aşteptat, ceea ce se realizează prin minimizarea dispersiei beneficiului total

$$D^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j.$$

Urmărind, concomitent, maximizarea lui M şi minimizarea lui D^2, un mod de a aborda problema ar fi acela de a cere maximizarea funcţiei — obiectiv

$$f(X) = M - D^2 = \sum_{i=1}^{n} m_i x_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j.$$

Cum matricea de covarianţă $(\sigma_{ij})_{\substack{1 \le i \le n \\ 1 \le j \le n}}$ este, totdeauna, pozitiv definită, rezultă că forma pătratică

$$-\sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j$$

este concavă.

Scanned with OKEN Scanner

Restricţiile ce trebuie impuse variabilelor x_i sînt, în mod natural

$$\sum_{i=1}^{n} x_i = S,$$

$$x_j \geqslant 0, \; i = 1, 2, \ldots, n.$$

S-a ajuns astfel la un program pătratic de maximizare a unei funcţii concave.

3.2. Caracterizarea soluţiilor problemei

Întrucît toate rezultatele teoretice stabilite pentru programarea convexă îşi păstrează valabilitatea, ne propunem să transcriem condiţiile Kuhn-Tucker pentru problema 1).

Avem

$$F(X, Y) = dX' + X'CX + Y'(AX - b)$$

(peste tot aici, X şi Y sînt utilizaţi ca vectori coloană)

$$\frac{\partial F}{\partial x_j} = d_j + 2\sum_{k=1}^{n} a_{kj}x_k + \sum_{k=1}^{n} a_{kj}y_k, \qquad j = 1, 2, \ldots, n,$$

$$\frac{\partial F}{\partial y_i} = \sum_{k=1}^{n} a_{ik}x_k - b_i. \qquad i = 1, 2, \ldots m$$

Notăm

$$V = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}, \quad U = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix}$$

cu $v_j = \frac{\partial F}{\partial x_j}$, $(j = 1, 2, \ldots, n)$, $u_i = \frac{\partial F}{\partial y_i}$, $(i = 1, 2, \ldots, m)$.

Atunci

$$V = d' + 2CX + A'Y,$$

(2)

$$U = -AX + b.$$

Scanned with OKEN Scanner

Barankin şi Dorfman au dat următoarea formă, relaţiilor Kuhn-Tucker, aplicate programării pătratice:

$$AX + U = b, \tag{3}$$

$$2CX - V + A'Y = -d' \tag{4}$$

$$X \geqslant 0,\ Y \geqslant 0,\ V \geqslant 0,\ U \geqslant 0, \tag{5}$$

$$X'Y + Y'U = 0. \tag{6}$$

Primele două egalităţi rezultă din (2), celelalte reprezintă transcrierea condiţiilor (13) — (14) din § 2.

În această formulare specifică, condiţiile Kuhn-Tucker se enunţă astfel:

X^* este soluţie optimă a programului (1) *dacă şi numai dacă există vectorii* Y^*, $U^* \in R^m$, $V^* \in R^n$ *astfel că relaţiile* (3)—(6) *să fie satisfăcute.*

Ţinînd seama de (5), egalitatea (6), spune că pentru fiecare ($j = 1, 2, \ldots, n$) şi pentru fiecare i, ($i = 1, 2, \ldots, m$) cel puţin una din perechile x_j, v_j, respectiv y_i, u_i este nulă. Ca atare, o soluţie optimă implică o soluţie X^*, V^*, Y^*, U^* (cu $2(n + m)$ componente nenegative) a sistemului liniar de $n + m$ ecuaţii (3), (4), în care cel mult $n + m$ componente sînt nenule.

Adică o soluţie optimă este atinsă pentru o soluţie de bază a sistemului liniar (3), (4).

Această observaţie motivează posibilitatea adaptării metodei simplex pentru rezolvarea programelor pătratice.

3.3. Rezolvarea programelor, pătratice. Metoda lui Frank şi Wolfe

Din cele de mai sus, se desprinde ideia potrivit căreia rezolvarea programului (1) se reduce la determinarea unei soluţii X^*, V^*, Y^*, U^* a sistemului (3)—(6), sistem care poate fi adus la o formă mai concentrată, utilizînd notaţile

$$Z' = (X', U', V', Y') =$$

$$= (x_1, x_2, \ldots, x_n;\ u_1, u_2, \ldots, u_m;\ v_1, v_2, \ldots, v_n;\ y_1, y_2, \ldots, y_m)$$

$$\overline{Z}' = (V', Y', X', U') =$$

$$= (v_1, v_2, \ldots, v_n;\ y_1, y_2, \ldots, y_m;\ x_1, x_2, \ldots, x_n;\ u_1, u_2, \ldots, u_m).$$

Scanned with OKEN Scanner

Noua formă a sistemului va fi:

(7)

$$\left(\begin{array}{c|c|c|c} A & E_m & O & O \\ \hline 2C & O & E_n & A \end{array}\right) Z = \left(\begin{array}{c} b \\ \hline -d' \end{array}\right)$$

(am notat E_m, E_n, matricele unitate m, respectiv n-dimensionale);

(8) $$Z \geqslant 0;$$

(9) $$Z'\bar{Z} = 0.$$

Metoda propusă de Frank şi Wolfe caută vectorul Z care minimizează expresia

$$T(Z) = Z'\bar{Z}$$

în condiţiile (7), (8), ştiind că minimul este zero şi că este atins pentru o soluţie de bază a sistemului (7).

Iată, pe scurt, etapele algoritmului:

În prima etapă se determină o soluţie de bază Z^0 a sistemului (7), (8), folosind algoritmul simplex.

Dacă $T(Z^0) = 0$, atunci componenţa X^0 a lui Z^0 este soluţia optimă a problemei (1).

Altminteri, se îmbunătăţeşte soluţia, printr-un şir de iteraţi construit astfel:

Fie, după k etape, Z^k soluţia de bază obţinută şi W^k o altă soluţie posibilă (nu necesar de bază) a sistemului (7), (8) (pentru prima iteraţie se poate lua $W^0 = Z^0$).

Se aplică algoritmul simplex pentru minimizarea funcţionalei liniare

$$L(Z) = Z'\bar{W}^k.$$

plecînd de la Z^k ca soluţie de bază iniţială.

Fie

$$Z_0^k = Z^k, \quad Z_1^k, \; Z_2^k, \; \ldots, \; Z_l^k, \; \ldots$$

şirul soluţiilor de bază obţinute prin aplicarea acestui algoritm.

Întrerupem procesul dacă

a) $T(Z_l^k) = (Z_l^k)' \cdot Z_l^k = 0$

sau dacă

$$\text{b)}\ L(Z_l^k) = (Z_l^k)' \cdot \overline{W}^k \leqslant \frac{1}{2}(W^k)' \ \overline{W}^k = \frac{1}{2} T(W^k).$$

În cazul a), deoarece Z_l^k este o soluție de bază pentru care T ia valoarea 0, ea va da, implicit, soluția optimă a programului.

În cazul b), inițiem o nouă iterație, punînd

$$Z^{k+1} = Z_l^k,$$

$$W^{k+1} = W^k + \mu(Z^{k+1} - W^k)$$

$$\text{cu } \mu = \min\left\{1, \frac{(W^k - Z^{k+1}) \cdot \overline{W}^k}{(Z^{k+1} - W^k)' \cdot (\overline{Z}^{k+1} - \overline{W}^k)}\right\}.$$

Se poate demonstra, că algoritmul dă în mod necesar, soluția optimă după un număr finit de pași.

Vom ilustra, în continuare, metoda descrisă pe un exemplu numeric.

Fie problema (aceeași din § 2):

Să se minimizeze

$$f(X) = -x_1 - 2x_2 + \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2$$

în condițiile

$$\begin{aligned} 2x_1 + 3x_2 &\leqslant 6, \\ x_1 + 4x_2 &\leqslant 5, \\ x_1 &\geqslant 0, \\ x_2 &\geqslant 0 \end{aligned}$$

(am înlocuit, condiția de maxim cu una de minim multiplicînd cu -1 funcția — obiectiv).

În conformitate cu notațiile standard

$$d = (-1, -2), \qquad C = \begin{Vmatrix} 1/2 & 0 \\ 0 & 1/2 \end{Vmatrix},$$

$$A = \begin{Vmatrix} 2 & 3 \\ 1 & 4 \end{Vmatrix}, \qquad b = \begin{pmatrix} 6 \\ 5 \end{pmatrix}.$$

Scanned with OKEN Scanner

Evident C este pozitiv definită (de aici unicitatea optimului):

$$B = \begin{Vmatrix} 2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 2 & 1 \\ 0 & 1 & 0 & 0 & 0 & -1 & 3 & 4 \end{Vmatrix}$$ (am notat cu B matricea din (7));

$$g = \left(\frac{b}{d'}\right) = \begin{pmatrix} 6 \\ 5 \\ 1 \\ 2 \end{pmatrix};$$

$Z' = (x_1, x_2, u_1, u_2, v_1, v_2, y_1, y_2) = (z_1, z_2, \ldots, z_8)$.
Formulăm problema

$$\min\{T(Z) = Z'\overline{Z}\},$$
$$BZ = g,$$
$$Z \geqslant 0.$$

Pentru a iniţia algoritmul simplex introducem variabilele auxiliare t_1, t_2 astfel ca

$$BZ + FT = g,$$
$$Z \geqslant 0,\ T \geqslant 0,$$

$$\text{cu } T = \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}, \qquad F = \begin{Vmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{Vmatrix}.$$

Acum în matricea $(B\vdots F)$ a sistemului putem remarca existenţa vectorilor unitari.

Cu elementele problemei, completăm primul tabel simplex (tabelul 32) cu mici deosebiri de notaţie faţă de modul consacrat în capitolul II, dictate de necesităţile unei mai mari clarităţi.

Ultimul tabel simplex dă soluţia de bază.

$$(Z^0)' = (0, 0, 6\ ,5, 0, 0, 2/5\ ,\ 1/5)$$

(se putea obţine, rezolvînd sistemul pătrat corespunzător coloanelor a doua, a patra, a şaptea şi a opta a matricei B).

Scanned with OKEN Scanner

Tabelul 32

C_B	c_j	$\rightarrow$	0	0	0	0	0	0	0	0	α	α
$\downarrow$	variabile bazice	g	x_1	x_2	u_1	u_2	v_1	v_2	y_1	y_2	e_1	e_2
0	u_1	6	2	3	1	0	0	0	0	0	0	0
0	u_2	5	1	4	0	1	0	0	0	0	0	0
α	$\leftarrow t_1$	1	1	0	0	0	−1	0	$\boxed{2}$	1	1	0
α	t_2	2	0	1	0	0	0	−1	3	4	0	1
	$z_j - c_j$	3α	α	α	0	0	$-\alpha$	$-\alpha$	5α	5α	α	α
0	u_1	6	2	3	1	0	0	0	0	0	0	
0	u_2	5	1	4	0	1	0	0	0	0	0	
0	y_1	1/2	1/2	0	0	0	−1/2	0	1	1/2	0	
α	$\leftarrow t_2$	1/2	−3/2	1	0	0	3/2	−1	0	$\boxed{5/2}$	1	
	$z_j - c_j$	1/2	$\frac{3}{2}\alpha$	$-\alpha$	0	0	$\frac{3}{2}\alpha$	$-\alpha$	0	$\frac{5}{2}\alpha$	α	
0	u_1	6	2	3	1	0	0	0	0	0		
0	u_2	5	1	4	0	1	0	0	0	0		
0	y_1	2/5	4/5	−1/5	0	0	−4/5	1/5	1	0		
0	y_2	1/5	−3/5	2/5	0	0	3/5	−2/5	0	1		

Luăm $W^0 = Z^0$

$$T(Z^0) = (0, 0, 6, 5, 0, 0, 2/5, 1/5) \begin{pmatrix} 0 \\ 0 \\ 2/5 \\ 1/5 \\ 0 \\ 0 \\ 6 \\ 5 \end{pmatrix} = 34/5 > 0.$$

Scanned with OKEN Scanner

Căutăm să minimizăm forma liniară

$$L(Z) = Z'\overline{W}^0 = \frac{2}{5}u_1 + \frac{1}{5}u_2 + 6y_1 + 5y_2.$$

În continuare, la fiecare pas, valoarea lui L, dată de tabloul simplex, va fi comparată cu

$$\frac{1}{2}T(W^0) = \frac{17}{5}.$$

Etapa a doua (determinarea lui Z')

Se pleacă de la ultimul tabel simplex din 32 și se obține tabelul 33.

După primele două iterate valorile lui L fiind 34/5, respectiv, 27/5, deci, inferioare lui $\frac{1}{2}T(W^0)$, algoritmul a fost continuat.

La sfîrșitul părții a treia a tabelului 33 oprim algoritmul deoarece $(Z_2^0)'Z_2^0 = 0$.

Ca atare, conform cu a), componentele

$x_1^* = 13/17$, $x_2^* = 18/17$ ale ultimei soluții de bază reprezintă optimul căutat.

3.4. Dualitatea în programarea pătratică

Ca și pentru problemele de programare liniară, putem formula pentru un program pătratic un program dual asociat.

Teorema dualității capătă o formă mai slabă, comparativ cu cazul liniar, însă păstrează totuși un conținut destul de bogat.

Dacă problema primală este

$$\min \{f(X) = dX + X'CX | AX \leqslant b\}$$

atunci problema duală va fi

$$\max \left\{ f'(X) = dX + X'CX + Y'(AX - b) \,\middle|\, \frac{\partial f'}{\partial x_j}(X, Y) \geqslant 0, \right.$$

$$\left. j = 1, \ldots, n, Y \geqslant 0 \right\}$$

sau, ținînd seama de notațiile anterioare

$$\max \{- X'CX - b'Y | 2CX + A'Y = -d', \quad Y \geqslant 0\}.$$

Scanned with OKEN Scanner

Tabelul 33

C_B	c_j	→	0	0	2/5	1/5	0	0	6	5	
↓	variabile bazice	g	x_1	x_2	u_1	u_2	v_1	v_2	y_1	y_2	$Z'Z$
2/5	u_1	6	2	3	1	0	0	0	0	0	34/5
1/5	u_2	5	1	4	0	1	0	0	0	0	
6	←y_1	2/5	$\boxed{4/5}$	−1/5	0	0	− 4/7	1/5	1	0	
5	y_2	1/5	−3/5	2/5	0	0	3/5	−2/5	0	1	
	$z_j - c_j$	L=34/5	14/5	14/5	0	0	− 9/5	−4/5	0	0	
2/5	u_1	5	0	7/2	1	0	2	−1/2	−5/2	0	9/2
1/5	←u_2	9/2	0	$\boxed{17/4}$	0	1	1	−1/4	−5/4	0	
0	x_1	1/2	1	−1/4	0	0	−1	1/4	5/4	0	
5	y_2	1/2	0	1/4	0	0	0	−1/4	3/4	1	
	$z_j - c_j$	$L(Z_1^0)$= 27/5	0	7/2	0	0	1	−3/2	−7/2	0	
2/5	u_1	22/17	0	0	1	−14/17	20/17	−5/17	−25/17	0	0
0	x_2	18/17	0	1	0	4/17	4/17	−1/17	−5/17	0	
0	x_1	13/17	1	0	0	1/17	−16/17	4/17	20/17	0	
5	y_2	4/17	0	0	0	−1/17	−1/17	−4/17	14/17	1	
	$z_j - c_j$	$L(Z_2^0)$= =144/85	0	0	0	−14/17	3/17	−22/17	−60/17	0	

Scanned with OKEN Scanner

Teorema dualității. *Dacă mulțimea punctelor admisibile pentru una din probleme este vidă, atunci la fel va fi și pentru cea de a doua problemă, sau cea de a doua problemă nu are soluție finită. Dacă pentru una din probleme soluția nu este finită, atunci pentru cealaltă nu există puncte admisibile. Dacă una din probleme are soluție optimă, atunci și cealaltă problemă admite de asemenea soluție optimă și valorile optime corespunzătoare ale celor două funcții obiectiv coincid.*

Dacă X^* este soluție optimă a problemei primale, atunci există Y^* astfel ca (X^*, Y^*) să fie soluție optimă a dualei. Reciproc, dacă C este pozitiv definită, și (X^*, Y^*) este soluție optimă pentru problema duală, atunci X^* este soluție optimă pentru problema primală.

Scanned with OKEN Scanner

Capitolul IV

PROGRAMAREA DISCRETĂ

§ 1. PROBLEME TIPICE APARȚINÎND PROGRAMĂRII DISCRETE

1.1. Exemple

O serie de situații practice de mare importanță conduc la probleme de programare matematică de o factură deosebită de cele întîlnite în capitolele precedente. Anume natura concretă a fenomenelor modelate impune cerința ca variabilele supuse restricțiilor problemei să nu poată lua decît valori din anumite mulțimi discrete (cel mai frecvent numere întregi). Asemenea probleme, în care toate variabilele, sau numai o parte din ele, sînt obligate să varieze în mulțimi discrete, bine determinate, vor face obiectul studiului acestui capitol.

Vom începe prin a prezenta două exemple concrete:

Exemplul 1°. Să presupunem că un vapor trebuie să fie încărcat cu mărfuri de diferite sorturi în limita capacității sale C. Presupunînd că se au în vedere n mărfuri diferite, caracterizate fiecare, printr-o greutate și printr-o valoare specifică, se pune problema determinării modului optim în care trebuie aleasă încărcătura, astfel ca valoarea totală a mărfurilor transportate să fie maximă.

Fie v_i, m_i, valoarea și, respectiv, masa unei unități din tipul i de marfă $(i = 1, 2, \ldots, n)$.

Notînd cu $x_i (i = 1, 2, \ldots, n)$ numărul de unități de marfă de tipul i ce va fi încărcat, problema poate fi formulată astfel:

I. Să se maximizeze

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} v_i x_i,$$

în condițiile

$$\text{II.} \quad \sum_{i=1}^{n} m_i x_i \leqslant C,$$

Scanned with OKEN Scanner

III. $x_i = 0, 1, 2, \ldots \qquad i = 1, 2, \ldots, n.$

Ultima condiție a fost impusă de faptul că mărfurile se prezintă în unitățile presupuse indivizibile.

Exemplul 2°. Fie n localități diferite și fie d_{ij} distanțele dintre localitățile i și j ($i, j = 1, 2, \ldots, n$).

Să se stabilească ruta optimă pentru o cursă de avioane, care să treacă prin fiecare din cele n localități o singură dată (cu punct de plecare și de sosire o aceiași localitate), avînd în vedere minimizarea parcursului total (se presupune că avionul zboară în linie dreaptă între două localități consecutive ale rutei).

Din examinarea sumară a datelor problemei, se poate face remarca că orice rută posibilă conține n etape, reprezentînd, fiecare, un zbor între două localități.

Pentru formularea matematică a problemei fie x_{ijt} indicatorul căruia îi atribuim valoarea 1, dacă etapa t a rutei recomandă zborul din localitatea i spre localitatea j, și valoarea 0 în rest ($i, j, t = 1, 2, \ldots, n$).

Putem scrie imediat următoarele relații:

$$\text{I.} \quad \sum_{j=1}^{n} \sum_{t=1}^{n} x_{ijt} = 1, \qquad i = 1, 2, \ldots, n,$$

deoarece, dintr-o localitate i avionul pleacă o singură dată către o singură altă localitate;

$$\text{II.} \quad \sum_{i=1}^{n} \sum_{t=1}^{n} x_{ijt} = 1, \qquad j = 1, 2, \ldots, n,$$

relație asemănătoare cu precedenta, exprimînd faptul că fiecare localitate j este punctul de sosire al avionului într-o singură etapă;

$$\text{III.} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ijt} = 1, \qquad t = 1, 2, \ldots, n,$$

deoarece fiecare etapă t a rutei este formată dintr-un singur zbor de la o localitate la o alta;

$$\text{IV.} \quad \sum_{i=1}^{n} x_{ijt} = \sum_{i=1}^{n} x_{ijt+1}, \qquad j, t = 1, 2, \ldots, n.$$

Scanned with OKEN Scanner

Această ultimă relație exprimă faptul că dacă, la o etapă t dată, localitatea j a constituit punctul de sosire, atunci la următoarea etapă j va fi punct de plecare (cu notația $x_{ijn+1} = x_{ij1}$, relația este valabilă și pentru punctul terminus al rutei).

În sfîrșit, din însăși definiția indicatorului x_{ijt}, rezultă

V. $x_{ijt} = 0$ sau 1, $\qquad i, j, t = 1, 2, \ldots, n.$

Orice sistem x_{ijt} $(i, j, t = 1, 2, \ldots, n)$, satisfăcînd I—V, descrie complet și în mod unic o rută posibilă a avionului. Lungimea totală a acestei rute va fi

$$D = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{t=1}^{n} d_{ij} x_{ijt}.$$

În concluzie, putem da acestei probleme următorul enunț:

Să se determine variabilele x_{ijt}, care în condițiile I — V realizează

VI. minim D.

Restricțiile privind discontinuitatea variabilelor sînt de fapt proprii multor modele economice. Chiar în problemele de planificare enunțate în legătură cu programarea liniară, era foarte natural să se presupună că valorile x_i, reprezentînd numărul de unități de produs i planificat, trebuie să ia valori întregi.

În contextul unor probleme însă, asemenea condiții pot fi cu ușurință ignorate. Motivul: valorile variabilelor fiind relativ, mari o rotunjire a soluției optime obținute în programul continuu (care presupune pentru variabile posibilitatea de a lua orice valoare reală verificînd sistemul de inegalități asociat) ne oferă un plan de acțiune convenabil, fracțiunile adăugate sau omise, nemodificînd sensibil valoarea funcției obiectiv.

Cu totul alta este situația în problemele în care, soluțiile sînt exprimate prin numere mici, comparabile cu unitatea.

Un exemplu numeric ilustrează dezacordul sensibil, dintre soluția obținută pe calea descrisă mai sus și soluția reală a problemei pentru care și restricțiile de tip discret au fost luate în considerare.

Exemplul 3°. Să se rezolve problema:

În condițiile

$$\text{I.} \quad \begin{aligned} 0{,}5x_1 + x_2 &\leqslant 1{,}75, \\ x_1 + 0{,}3x_2 &\leqslant 1{,}5 \end{aligned}$$

Scanned with OKEN Scanner

II. $x_1 \geqslant 0,\ x_2 \geqslant 0.$

III. $x_1,\ x_2$ întregi,

Să se determine maximul funcţiei

IV. $f(x_1, x_2) = 0{,}25x_1 + x_2.$

În figura 17, domeniul haşurat $OABC$ reprezintă mulţimea punctelor (x_1, x_2) satisfăcînd I şi II.

Problema I, II, IV, obţinută prin omiterea condiţiilor III, ne indică drept soluţie optimă.

$x_1^0 = 0,\ x_2^0 = 1{,}75$

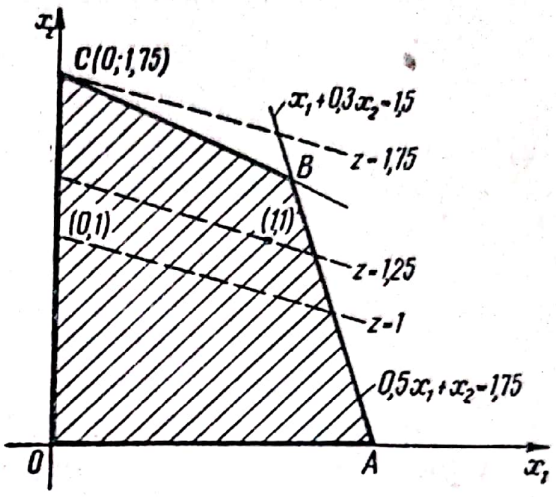


Fig. 17.

pentru care

$$f(x_1^0, x_2^0) = 1{,}75.$$

Ţinînd seama acum de III soluţia trebuie modificată. Procedeul obişnuit, de rotunjire în mulţimea soluţiilor posibile, ne-ar indica

$$x_1 = 0, \quad x_2 = 1,$$

pentru care

$$f(x_1, x_2) = 1.$$

Putem însă remarca, că în domeniul $OABC$ singurele puncte care satisfac şi condiţia III, au coordonatele: (0, 0), (1, 0), (1, 1), (0, 1).

Or, dintre acestea, cel de al treilea reprezintă soluţia optimă, maximizînd funcţia f (în raport cu celelalte trei puncte). Soluţia optimă-obţinută aici prin simpla inspectare a celor patru soluţii posibile este

$$x_1^* = 1,\ x_2^* = 1,$$

pentru care

$$f(x_1^*, x_2^*) = 1{,}25.$$

Scanned with OKEN Scanner

Diferența dintre valoarea reală de optim și cea rotunjită este sensibilă.

Caracterul discret al unui program de tipul considerat aici, conferă proprietatea de finitudine — în majoritatea cazurilor — mulțimii soluțiilor. S-ar putea crede atunci că orice program discret se poate rezolva, ca în exemplul 3° prin inspectarea tuturor situațiilor posibile.

Nu este greu de observat că această cale este total neconvenabilă. În problema din exemplu 2°, numărul total al rutelor posibile este $(n-1)!$. Chiar pentru valori moderate ale lui n, numărul situațiilor posibile ar deveni, totuși mult prea mare pentru metodele combinatoriale uzuale.

S-a pus astfel problema creerii unor algoritmi speciali pentru rezolvarea programelor discrete.

1.2. Cazuri de mare interes

Deoarece majoritatea problemelor de programare discretă, care se întîlnesc în practică, impun cerința ca variabilele să fie întregi ne vom limita, în cele ce urmează, numai la acest tip (cunoscut și sub numele de programare în numere întregi).

Aceasta nu înseamnă, totuși o restrîngere a generalității, deoarece și cazurile generale se pot aduce, fără dificultate, la forma susmenționată.

De exemplu, dacă variabila x_i, poate lua numai un număr finit de valori $k_{i1}, k_{i2}, k_{i2}, \ldots, k_{iN}$, se poate scrie:

$$x_i = \delta_{i1}k_{i1} + \delta_{i2}\,k_{i2} + \ldots + \delta_{iN}k_{iN}$$

cu $\sum_{j=1}^{N} \delta_{ij} = 1$, $\delta_{ij} = 0$ sau 1, obținîndu-se, pe această cale un program în variabilele întregi δ_{ij}.

Cum uneori, în programele discrete, numai asupra unei părți din mulțimea variabilelor acționează restricțiile adiționale de discontinuitate, celelalte variabile păstrîndu-și caracterul continuu, vom face distincție între două tipuri fundamentale: programe total discrete — în care toate variabilele trebuie să ia valori discrete — și programe mixte — în care unele variabile au caracter discret, celelalte variind într-un domeniu continuu.

Scanned with OKEN Scanner

Orice problemă de programare discretă este, prin esența ei, un program neliniar. Chiar dacă funcția obiectiv și sistemul de restricții sînt exprimate prin forme liniare, datorită caracterului discret al variabilor, ele nu mai sînt — matematic vorbind — funcții liniare.

Vom folosi, totuși termenul de „programare liniară discretă" pentru astfel de probleme.

În conformitate cu clasificarea făcută anterior vom enunța două tipuri de programe liniare discrete.

Forma generală a unei probleme de programare liniară total discretă este următoarea:

(1) Să se determine

$$\text{maxim (minim)} \{f(X) = CX = c_1x_1 + c_2x_2 + \ldots + c_nx_n\}$$

în condițiile

$$(2) \qquad AX\{\geqslant ; =, \leqslant\} b,$$

$$X \geqslant 0,$$

(3) x_i întregi, $\qquad i = 1, 2, \ldots, n,$

unde A este o matrice $(m \times n)$ și b un vector coloană de dimensiune m.

O problemă de programare mixtă (liniară) poate fi enunțată prin:

(1) Să se determine

maxim (minim) $f(X)$

în condițiile

$$(5) \quad AX\{\geqslant, =, \leqslant\} b,$$

$$X \geqslant 0;$$

(6) x_i întregi, $\qquad i \in I_1 \subset \{1, 2, \ldots, n\}$

Programarea liniară — total discretă sau mixtă — face posibilă soluționarea a numeroase probleme care nu se pot încadra în limitele programării liniare obișnuite, și pentru care nu se cunosc, pînă în prezent, metode eficiente de rezolvare.

Scanned with OKEN Scanner

Este vorba de tipul de programe în care apar restricţii sau grupe de restricţii ce se exclud reciproc, sau care nu sînt satisfăcute, obligatoriu, în totalitate.

Să presupunem că sistemul de restricţii al unui program, cu funcţia obiectiv liniară este formulat astfel:

(7) „Una şi numai una din următoarele inegalităţi, este adevărată"

$$g_1(X) \geqslant 0,$$
$$g_2(X) \geqslant 0,$$
$$\ldots\ldots$$
$$g_m(X) \geqslant 0,$$

cu $g_i(X) = \sum_{k=1}^{n} a_{ik}x_k - b_i, \qquad i = 1, 2, \ldots, m.$

Dacă, în plus, presupunem că variabilele problemei sînt mărginite, fie L_i marginea inferioară a funcţiei $g_i (i = 1, 2, \ldots, m)$.

Inegalităţile $g_i(X) \geqslant L_i$, fiind banal verificate, nu implică o modificare a condiţiilor problemei.

Este uşor de verificat că sistemul de restricţii 7) este echivalent cu:

$$g_i(X) - (1 - \delta_i)L_i \geqslant 0, \qquad i = 1, 2, \ldots, m;$$

$$\sum_{i=1}^{m} \delta_i = 1;$$

$$\delta_i = 0 \text{ sau } 1, \qquad i = 1, 2, \ldots, m,$$

şi astfel, problema iniţială devine o problemă de programare liniară mixtă, în care variabilele $x_1, x_2, \ldots, x_n$, sînt continue, iar variabilele $\delta_1, \delta_2, \ldots, \delta_m$, pot lua doar valorile întregi 0 sau 1.

Un caz ceva mai general decît cel descris mai sus, îl prezintă programul în care restricţiile cer ca un număr k de inegalităţi dintr-un sistem de m inecuaţii date să fie verficate.

Fie acesta

(8) $$g_i(X) \geqslant 0, \qquad i = 1, 2, \ldots, m.$$

Scanned with OKEN Scanner

Dacă L_i are aceiaşi semnificaţie ca mai înainte, atunci, se ajunge uşor la un program liniar mixt echivalent, în care restricţiile sînt

$$g_i(X) - \delta_i L_i \geqslant 0, \qquad i = 1, 2, \ldots, m;$$

$$\sum_{i=1}^{m} \delta_i = m - k;$$

$$\delta_i = 0 \text{ sau } 1, \qquad i = 1, 2, \ldots, m.$$

O variantă a ultimei probleme ar putea fi dată prin enunţarea următoarului sistem de restricţii:

„Cel puţin k inegalităţi din (8) să fie satisfăcute"

Echivalentul liniar al acestui sistem de restricţii ar fi

$$g_i(X) - \delta_i L_i \geqslant 0, \qquad i = 1, 2, \ldots, m;$$

$$\sum_{i=1}^{m} \delta_i \leqslant m - k;$$

$$\delta_i = 0 \text{ sau } 1, \qquad i = 1, 2, \ldots, m.$$

În toate cazurile discutate, mulţimile soluţiilor posibile se prezintă ca reuniuni de domenii convexe, care, în general, nu formează ele însile domenii convexe, aşa cum pretinde un program liniar propriu-zis. Prin transformările operate, dacă funcţia obiectiv este liniară, toate aceste programe conduc la programe liniare mixte.

O altă clasă de probleme ale programării matematice care pot fi abordate în cadrul teoriei programării discrete, o formează programele neliniare continue, în care metodele de rezolvare proprii acestui domeniu nu sînt eficiente.

Vom presupune că un program conţine restricţii neliniare de forma

$$(9) \qquad g_i(X) \{\geqslant, =, \leqslant\} 0, \qquad i = 1, 2, \ldots, m,$$

unde $g_i(X) = \sum_{j=1}^{n} g_{ij}(x_j)$, g_{ij} fiind funcţii oarecare de o singură variabilă x_j.

Scanned with OKEN Scanner

Ideia care stă la baza următoarei transformării este aproximarea fiecărei g_{ij} printr-o linie poligonală, înlocuind-o astfel printr-o formă liniară.

Pentru aceasta fie a_{jk} valori, convenabil alese ale variabilei x_j, $(k = 1, 2, \ldots, N_j$ pentru fiecare $j = 1, 2, \ldots, n)$ şi fie funcţia liniară

$$h_{ij}(\delta_{j1}, \delta_{j2}, \ldots, \delta_{jNj}) = \sum_{k=1}^{Nj} \delta_{jk} g_{ij}(a_{jk}), \quad i = 1, 2, \ldots, m,$$

$j = 1, 2, \ldots, n.$

Operînd şi în expresia funcţiei obiectiv înlocuirea

$$x_j = \sum_{k=1}^{Nj} \delta_{jk} a_{jk},$$

se obţine o problemă de programare total discontinuă, în care sistemul de restricţii iniţial (9) este aproximat prin unul liniar

$$\sum_{j=1}^{n} h_{ij}(\delta_{j1}, \delta_{j2}, \ldots, \delta_{jNj}) \{ \leqslant , = , \geqslant) 0,$$

$$\sum_{k=1}^{Nj} \delta_{jk} = 1 \tag{9'}$$

$$\delta_{jk} = 0 \text{ sau } 1.$$

Fără a intra în amănunte, să notăm faptul că, pentru orice problemă de optimizare a unei funcţii separabile de forma

$$f(X) = \sum_{j=1}^{n} f_j(x_j),$$

se poate obţine un program discret aproximativ — echivalent, prin înlocuirea funcţiilor f_j cu forme liniare şi introducerea în program a unor variabile discrete, într-o manieră asemănătoare cu cea folosită în cazul discutat anterior.

În cele ce urmează vom trata mai pe larg problemele de programare liniară discretă, dat fiind importanţa lor practică, şi, avînd totodată în vedere şi existenţa unor algoritmi de rezolvare pentru acest tip. De asemenea, ca tip reprezentativ de problemă discretă neliniară, vom aborda şi programarea discretă pătrată.

Scanned with OKEN Scanner

§ 2. ALGORITMUL LUI GOMORY (CICLIC) PENTRU PROGRAMAREA LINIARĂ TOTAL DISCRETĂ

2.1. Descrierea algoritmului

Presupunem programul adus la forma

(1) $AX = b$ (A de ordin $(m \times n)$, cu rangul m),

(2) $X \geqslant 0$,

(3) x_i întregi, $\quad i = 1, 2, \ldots, n.$

(4) Să se maximizeze

$$f(X) = CX.$$

Metoda propusă de Gomory pentru rezolvarea unui astfel de program, constă în rezolvarea succesivă a unui şir de programe liniare obişnuite (cu variabile continui) obţinute din (1), (2), (4), prin adăugarea cîte unei relaţii suplimentare.

Construcţia acestor relaţii adiţionale se face avînd în vedere relaţiile (3), fapt care determină în final, obţinerea unei soluţii cu toate componentele întregi.

Iată, pe scurt, descrierea algoritmului:

Într-o primă etapă preliminară se rezolvă problema de programare ordinară (1), (2), (4).

Fie $X^0 = (x_1^0, x_2^0, \ldots, x_n^0)$ soluţia optimă. Dacă toţi x_i^0 sînt întregi, atunci X^0 constitue, totodată, optimul, programului discret dat.

Să presupunem că acest lucru nu se întîmplă. Fără a restrînge generalitatea, putem presupune că primele m variabile sînt bazice pentru soluţia optimă a programului liniar considerat şi fie tabelul 34 corespunzător ultimei iterate a algoritmului simplex aplicat acestei probleme.

Fie x_i^0 prima componentă a soluţiei (în ordinea crescătoare a indicelui), neîntreagă.

Punînd în evidenţă părţile întregi ale componentelor liniei i, acestea se pot scrie

(5) $\quad x_i^0 = [x_i^0] + r_{i0}, \quad z_{ik} = [z_{ik}] + r_{ik}, \; k = m + 1, \ldots, n,$

Tabelul 34

C_B	$c_j \rightarrow$		c_1	...	c_i	...	c_m	c_{m+1}	...	c_n
↓	variabile bazice	P_0	P_1	...	P_i	...	P_m	P_{m+1}	...	P_n
c_1	x_1	x_1^0	1	...	0	...	0	z_{1m+1}	...	z_{1n}
	...	...	...	...	...	...	...	...	...	...
c_i	x_i	x_i^0	0	...	1	...	0	z_{im+1}	...	z_{in}
...	...	...	...	...	...	...	...	...	...	...
c_m	x_m	x_m^0	0	...	0	...	1	z_{mm+1}	...	z_{mn}

Evident,

(6) $\quad 0 \leqslant r_{i0} < 1, \quad 0 \leqslant r_{ik} < 1, \quad k = m+1, \ldots, n.$

Fiind componentă bazică a soluției, x_i^0 va satisface ecuația

$$x_i^0 = x_i + \sum_{k=m+1}^{n} z_{ik} x_k$$

sau, ținînd seama de (5),

$$[x_i^0] + r_{i0} = x_i + \sum_{k=m+1}^{n} [z_{ik}] x_k + \sum_{k=m+1}^{n} r_{ik} x_k.$$

De aici,

(7) $$r_{i0} - \sum_{k=m+1}^{n} r_{ik} x_k = x_i + \sum_{k=m+1}^{n} [z_{ik}] x_k - [x_i^0].$$

Dar, în virtutea relațiilor (2) și (6),

$$\sum_{k=m+1}^{n} r_{ik} x_k \geqslant 0.$$

Rezultă

(8) $$x_i + \sum_{k=m+1}^{n} [z_{ik}] x_k - [x_i^0] \leqslant r_{i0}.$$

Scanned with OKEN Scanner

Dacă ţinem seama acum şi de (3), putem afirma că membrul stîng din (8) trebuie să fie un număr întreg. Cum el trebuie să fie inferior numărului subunitar r_{i0}, rezultă

$$(9) \qquad x_i + \sum_{k=m+1}^{n} [z_{ik}]x_k - [x_i^0] = \text{întreg negativ.}$$

Atunci (9), (7), dau

$$(10) \qquad r_{i0} - \sum_{k=m+1}^{n} r_{ik}x_k = \text{întreg negativ.}$$

Introducînd o variabilă de compensare x_{n+1}, relaţia (10) se transformă în

$$(11) \qquad x_{n+1} - \sum_{k=m+1}^{n} r_{ik}x_k = -r_{i0}.$$

Din (10), x_{n+1} trebuie să aibă valori nenegative întregi.

Relaţia (11) constituie astfel o restricţie suplimentară, ce trebuie satisfăcută de variabilele programului discret şi de noua variabilă x_{n+1}. Remarcăm de asemenea, că aceasta din urmă este de acelaşi tip cu cele n variabile iniţiale ale problemei.

Primul ciclu al algoritmului propriu-zis:

Într-o primă etapă a ciclului, se construieşte relaţia (11) şi se completează tabelul 34, prin adăugarea liniei coeficienţilor din (11) şi a unei coloane pentru x_{n+1}.

Se obţine tabelul 35.

În noua soluţie a problemei, valoarea noii componente introduse este $-r_{i0}$, adică negativă.

Ca atare, este necesară eliminarea variabilei x_{n+1} din bază, ceea ce se realizează în a doua etapă a ciclului, aplicînd algoritmul simplex dual. Pentru acesta se ia ca punct de plecare tabelul 35.

Dacă etapa a doua nu elimină componentele negative din soluţie, programul dat nu are soluţie. Altfel, obţinem o nouă soluţie de bază X^1. Dacă X^1 are toate componentele întregi, va fi soluţia căutată în caz contrar se iniţiază un nou ciclu.

Oprim algoritmul în momentul în care soluţia este întreagă.

Scanned with OKEN Scanner

Tabelul 35

C_B	$c_j \rightarrow$		c_1	...	c_i	...	c_m	c_{m+1}	...	c_n	0
↓	variabile bazice	P_0	P_1	...	P_i	...	P_m	P_{m+1}	...	P_n	P_{n+1}
c_1	x_1	x_1^0	1	...	0	...	0	z_{1m+1}	...	z_{1n}	0
...	...	...	...	...	...	...	...	...	...	...	...
c_i	x_i	x_i^0	0	...	1	...	0	z_{im+1}	...	z_{in}	0
...	...	...	...	...	...	...	...	...	...	...	...
c_m	x_m	x_m^0	0	...	0	...	1	z_{mm+1}	...	z_{mn}	0
0	x_{n+1}	$x_{n+1}^0 = r_{i0}$	0	...	0	...	0	$-r_{im+1}$	...	$-r_{in}$	1

2.2. Convergența algoritmului

Din descrierea algoritmului rezultă că fiecare ciclu comportă adăugarea unei singure ecuații adiționale sistemului obținut la sfîrșitul etapei precedente și reoptimizarea noii probleme, astfel construită, prin algoritmul dual.

Deoarece procedeul durează pînă în momentul în care toate componentele soluției vor fi întregi, și cum fiecare nou ciclu presupune introducerea unei noi variabile, nu rezultă, în mod evident, finitudinea procesului, adică realizarea soluției optime (în întregi) după un număr finit de pași.

Se poate demonstra însă, că în anumite ipoteze suplimentare, procedeul este finit, conducîndu-ne la soluția optimă dacă aceasta există.

Ipotezele suplimentare cer ca toți coeficienții $c_i (i = 1, 2, \ldots, n)$ ai variabilelor din funcția obiectiv $f(X) = CX$ să fie întregi și ca domeniul $\mathcal{X}$ al soluțiilor posibile să fie mărginit (implicit și funcția f să fie mărginită).

Prima ipoteză nu este restrictivă deoarece, prin realegerea convenabilă a unității fizice de măsură, orice program liniar poate răspunde dezideratului formulat.

Scanned with OKEN Scanner

Pentru demonstrarea convergenței (într-un număr finit de pași) algoritmului, să introducem unele notații.

Fie P_j coloana j a matricei A și B matricea de bază corespunzătoare soluției optime a problemei (1), (2), (4).

Notăm cu Z_j ($j = 1, 2, \ldots, n$), Z_0, expresia vectorilor P_j, respectiv, $P_0 = b$ în baza B, și fie X_B vectorul coloană al variabilelor bazice.

Fie R mulțimea indicilor j corespunzători vectorilor P_j nebazici.

Avem

$$(12) \quad Z_0 = B^{-1}P_0,$$

$$Z_j = B^{-1}P_j, \qquad j = 1, 2, \ldots, n.$$

Fie de asemenea C_B vectorul linie al coeficienților funcției obiectiv, corespunzători variabilelor bazice. Atunci valoarea maximă a lui f este

$$z_0 = C_B Z_0.$$

Fie

$$S_0 = \begin{pmatrix} z_0 \\ Z_0 \\ 0 \end{pmatrix} (0 \text{ este matricea nulă de ordin card } (R)).$$

Algoritmul lui Gomory începe cu aceste elemente.

În ipotezele făcute, valoarea funcției f, pentru soluția optimă a programului discret, trebuie să fie întreagă. În consecință, dacă S_0 are toate componentele întregi pozitive, atunci $\begin{pmatrix} Z_0 \\ 0 \end{pmatrix}$ va reprezenta soluția optimă a problemei.

Altfel, plecînd de la $S_0(0) = S_0$, se construiește un șir:

$$S_0(0), \; S_0(1), \; \ldots, \; S_0(l), \; \ldots$$

fiecare termen al șirului fiind rezultatul, unui ciclu al algoritmului discret.

Cum fiecare ciclu comportă două etape, una din ele implicînd aplicarea algoritmului simplex dual, se poate întîmpla ca la o

Scanned with OKEN Scanner

anumită etapă, acesta să ne ducă la o situaţie inadmisibilă (imposibilitatea excluderii componentei negative din soluţie).

În acest caz, problema dată nu admite soluţie.

Să presupunem că remaximizarea problemei este posibilă la fiecare nou ciclu.

Vom arăta, că în acest caz, şirul $S_0(l)$ devine staţionar de la un anumit rang.

Dacă $z_0(l)$ reprezintă maximul funcţiei $f(X)$, obţinut la sfîrşitul ciclului l, acceptăm intuitiv, relaţia

$$z_0(l) \geqslant z_0(l+1), \qquad l = 0, 1, \ldots \tag{13}$$

Într-adevăr, $z_0(l+1)$ este maximul lui f obţinut în ciclul $l+1$, adică într-un sistem de restricţii mai tare (prin adăugarea unei restricţii în plus) decît cel din ciclul l, şi cum aceasta înseamnă restrîngerea mulţimii soluţiilor posibile, descreşterea maximului este evidentă.

Pe de altă parte, mărginirea lui f implică existenţa unei limite inferioare pentru şirul $z_0(l)$. Aceasta, împreună cu (13) atrage după sine convergenţa şirului.

Va exista atunci un număr întreg M, astfel ca

$$M < z_0(l) < M + 1, \tag{14}$$

pentru toate valorile lui l suficient de mari.

Să presupunem, prin absurd, că şirul $z_0(l)$ nu ar fi staţionar, adică

$$z_0(l) > z_0(l+1), \text{ pentru orice } l \in N.$$

Fie atunci, x_i^l componenta neîntreagă a vectorului X^l, soluţia ciclului l, pentru care se introduce noua restricţie

$$x_{n+1}^l = -r_{i0}^l + \sum_{k \in R^l} r_{ik}^l x_k \tag{15}$$

cu

$$r_{i0} = x_i^l - [x_i^l], \quad r_{ik}^l = z_{ik}^l - [z_{ik}^l]. \tag{16}$$

Pentru ca algoritmul simplex dual să elimine componenta negativă x_{n+1}^l din bază, este necesar să existe un indice q astfel ca $\frac{z_q^l - c_q}{r_{iq}^l} < 0$, $(z_q^l = C_{B^l} Z_q^l)$.

Scanned with OKEN Scanner

Atunci

$$(17) \qquad z_0(l+1) = z_0(l) + \frac{r_{i0}^l}{r_{iq}^l}(z_q^l - c_q) = z_0(l)\frac{r_{i0}^l}{r_{iq}^l}(c_q - z_q).$$

Ţinînd seama de (16) şi de ipoteza conform căreia coeficienţii c_k sînt întregi, rezultă

$$\frac{c_q - z_q^l}{r_{iq}^l} \geqslant 1,$$

şi atunci, din (17) avem

$$z_0(l+1) \leqslant z_0(l) - r_{i0}^l = z_0(l).$$

Or, pentru l suficient de mare, ţinînd seama de (14), ar rezulta

$$z_0(l+1) < M$$

ceea ce ar contrazice mărginirea lui f.

Înseamnă deci, că există un rang L_0, astfel ca

$$z_0(L_0) = z_0(L_0+1) + \ldots$$

Vom demonstra o proprietate asemănătoare şi pentru celelalte componente ale lui $S_0(l)$.

Fie x_j^l, componenta $j+1$ a vectorului $S_0(l)$.

Noua restricţie (15) şi algoritmul dual implică

$$(18) \qquad x_j^{l+1} = x_j^l + r_{i0}^l \frac{z_{jq}^l}{r_{iq}^l}.$$

De aici, deoarece $r_{iq}^l < 0$ şi $r_{i0}^l < z_{jq}^l$ rezultă

$$x_j^{l+1} < x_j^l - r_{i0}^l.$$

Mărginirea domeniului soluţiilor posibile face posibilă folosirea aceloraşi argumente, ca în demonstrarea stabilităţii şirului $z_0(l)$, pentru deducerea unui rezultat analog asupra şirului x_j^l.

Există deci L_j astfel ca toţi termenii şirului x_j^l să aibă aceeaşi valoare începînd de la rangul L_j.

Scanned with OKEN Scanner

De menționat că, x_j^l nu poate deveni stabil decît la o valoare întreagă.

Într-adevăr, dacă $x_{j+1}^l = x_j^l$, din (18) deducem

$$r_{i0}^l \frac{z_{jq}^l}{r_{iq}^l} = 0 \text{ și deci}$$

$$r_{i0}^l = 0, \text{ adică } x_j^l = [x_j^l]$$

În concluzie, va exista un rang $L = \max\{L_0, L_1, \ldots, L_n\}$, începînd de la care $S_0(L) = S_0(L+1) = \ldots$, ceea ce probează posibilitatea obținerii soluției optime după un număr finit de cicluri.

2.3. Un exemplu numeric

Fie programul total discret:

I. Să se maximizeze

$$f(X) = x_1 - 3x_2 + 3x_3.$$

În condițiile

II
$$\begin{aligned} 2x_1 + x_2 - x_3 + x_4 &= 4, \\ 4x_1 + 3x_2 + x_5 &= 2, \\ -3x_1 + 2x_2 + x_3 + x_6 &= 3. \end{aligned}$$

III. $X \geqslant 0$.

IV x_i întregi, $i = 1, 2, \ldots, 6$.

Etapa optimizatoare preliminară:

Rezolvăm problema $I - III$, folosind algoritmul simplex (tabelul 36).

Ultimul tablou indică soluția optimă neîntreagă.

Corespunzător componentei $x_1 = 1/2$ asociem o nouă ecuație.

Scanned with OKEN Scanner

Avem

$$r_{10} = 1/2,\ r_{12} = 1/4,\quad r_{15} = 1/4,\ r_{16} = 0.$$

Tabelul 36

C_B	c_j	→	1	−3	3	0	0	0
↓	variabile	P_0	P_1	P_2	P_3	P_4	P_5	P_6
0	x_4	4	2	1	−1	1	0	0
0	x_5	2	4	−3	0	0	1	0
0	←x_6	3	−3	2	[1]	0	0	1
	$c_j - z_j$	$z_0=0$	1	−3	3	0	0	0
0	x_4	7	−1	3	0	1	0	1
0	←x_5	2	[4]	−3	0	0	1	0
3	x_3	3	−3	2	1	0	0	1
	$c_j - z_j$	$z_0=9$	10	−9	0	0	0	−3
0	x_4	15/2	0	9/4	0	1	1/4	1
1	x_1	1/2	1	−3/4	0	0	1/4	0
3	x_3	9/2	0	−1/4	1	0	3/4	1
	$c_j - z_j$	$z_0=14$	0	−3/2	0	0	−5/2	−3

Deci

$$x_7 - \frac{1}{4}x_2 - \frac{1}{4}x_5 = -\frac{1}{2}$$

este prima relaţie adiţională.

Primul ciclu.

Completăm ultima parte a tabelului simplex 36 cu elementele corespunzătoare relaţiei nou introduse şi aplicăm algoritmul simplex dual pentru eliminarea componentei negative (tabelul 37).

Obţinem o nouă soluţie posibilă, care fiind întreagă, reprezintă soluţia programului discret.

Scanned with OKEN Scanner

Tabelul 37

C_B	c_j	→	1	−3	3	0	0	0	0
↓	varabile	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
0	x_4	15/2	0	9/4	0	1	1/4	1	0
1	x_1	1/2	1	−3/4	0	0	1/4	0	0
3	x_3	9/2	0	−1/4	1	0	3/4	1	0
0	←x_7	−1/2	0	**−1/4**	0	0	−1/4	0	1
	$c_j - z_j$	$z_0=14$	0	−3/2	0	0	−5/2	0	0
0	x_4	3	0	0	0	1	−81/4	1	9
1	x_1	2	1	0	0	0	1	0	−3
3	x_3	5	0	0	1	0	1	1	−1
−3	x_2	2	0	1	0	0	1	0	−4
	$c_j - z_j$	$z_0=11$	0	0	0	0	−1	−3	−6

Deci

$$x_1^* = 2,\ x_2^* = 2,\ x_3^* = 5,\ x_4^* = 3,\ x_5^* = x_6^* = 0,$$

și corespunzător,

$$f(X^*) = 11.$$

§ 3. REZOLVAREA PROGRAMELOR LINIARE MIXTE

3.1. Descrierea algoritmului

Principiul de rezolvare al unui program mixt este foarte asemănător cu cel care a stat la baza elaborării algoritmului ciclic pentru problemele total discrete.

Fie programul

$$AX = b,$$

$$X \geqslant 0,$$

x_i întregi pentru $i \in I\{1, 2, \ldots, n\}$;

maxim CX.

Scanned with OKEN Scanner

Cu notaţiile din § 2, dacă x_i^0, cu $i \in I$ este o componentă neîntreagă a soluţiei X^0, vom lua în considerare noua relaţie adiţională

$$(1) \qquad x_{n+1} - \sum_{k=m+1}^{n} d_{ik} x_k = -r_{i0}$$

unde

$$(2) \qquad d_{ik} = \begin{cases} z_{ik} & \text{dacă } z_{ik} \geqslant 0 \text{ şi } k \notin I, \\ \dfrac{r_{i0}}{1 - r_{i0}} |z_{ik}| & \text{dacă } z_{ik} < 0 \text{ şi } k \notin I, \\ z_{ik} & \text{dacă } k \in I \text{ şi } z_{ik} < r_{i0} \\ \dfrac{r_{i0}}{1 - r_{i0}} (1 - r_{ik}), & \text{dacă } k \in I \text{ şi } z_{ik} > r_{i0} \end{cases}$$

Cu aceste modificări, în construcţia restricţiilor adiţionale, algoritmul ciclic expus anterior se aplică şi în acest caz pînă la obţinerea unei soluţii optime cu toate componentele x_i, $i \in I$ întregi.

Finitudinea procesului a fost demonstrată folosind argumente asemănătoare celor din 2.2. în ipoteza restrictivă că valoarea optimă z^* a lui f trebuie să fie un număr întreg.

Înlăturarea acestei restricţii creează o serie de dificultăţi ce nu pot fi complet înlăturate prin introducerea unor ipoteze compensatoare mai slabe.

3.2. Un exemplu numeric

$$\frac{1}{2} x_1 + x_2 + x_3 = \frac{7}{4},$$

$$x_1 + \frac{3}{10} x_2 + x_4 = \frac{3}{2},$$

$$x_i \geqslant 0, \; i = 1, 2, 3;$$

x_1, x_2 întregi.

Să se maximizeze

$$f(X) = \frac{1}{4} x_1 + x_2.$$

Scanned with OKEN Scanner

Etapa preliminară optimizatoare constă în rezolvarea programului liniar în absența restricțiilor ce impun valori întregi variabilelor x_1 și x_2.

Prin aplicarea algoritmului simplex, obținem tabelul 38.

Tabelul 38

C_B	c_j	$\rightarrow$	1/4	1	0	0
$\downarrow$	variabile bazice	P_0	P_1	P_2 $\downarrow$	P_3	P_4
0	$\leftarrow x_3$	7/4	1/2	[1]	1	0
0	x_4	3/2	1	3/10	0	1
	$c_j - z_j$	$z_0 = 0$	1/4	1	0	0
1	x_2	7/4	1/2	1	1	0
0	x_4	39/40	17/20	0	−3/10	1
	$c_j - z_j$	$z_0 = 7/4$	− 1/4	0	−7/10	0

Soluția optimă astfel obținută este

$$x_1 = 0, \quad x_2 = 7/4, \quad x_3 = 0, \quad x_4 = 39/40.$$

Deoarece x_2 nu este număr întreg, corespunzător primei linii (cea a lui x_2), vom construi o nouă ecuație.

Avem, din (2),

$$d_{21} = 1/2\,; \; d_{23} = 1, \; r_{20} = 3/4.$$

Atunci, vom lua în considerație noua relație

$$x_5 - \frac{1}{2} x_1 - x_3 = -\frac{3}{4}\,.$$

Întocmim un nou tabel simplex (39), plecînd de la precedentul cu completările impuse de adăugarea noii restricții.

Tabelul 39

C_B	c_i	→	1/4	1	0	0	0
↓	variabile bazice	P_0	P_1	P_2	P_3	P_4	P_5
1	x_2	7/4	1/2	1	1	0	0
0	x_4	39/40	17/20	0	−3/10	1	0
0	← x_5	−3/4	|−1/2|	0	−1	0	1
	$c_j - z_j$	$z_0 = 7/4$	−1/4	0	−7/10	0	0
1	x_2	1	0	1	0	0	1
0	← x_4	−3/10	0	0	|−2|	1	17/10
1/4	x_1	3/2	1	0	2	0	−2
	$c_j - z_j$	$z_0 = 11/8$	0	0	−1/2	0	−1/2
1	x_2	1	0	1	0	0	1
0	x_3	3/20	0	0	1	−1/2	−17/20
1/4	x_1	6/5	1	0	0	1	− 3/10
	$c_j - z_j$	$z_0 = 13/10$	0	0	0	−1/4	−37/10

Au fost necesare două iterate ale algoritmului dual pentru eliminarea componentelor negative din soluţie.

Noua soluţie optimă, pentru problema modificată este:

$$x_1 = 6/5, \quad x_2 = 1, \quad x_3 = 3/20, \quad x_4 = 0.$$

Întrucît, componenta x_1 nu este întreagă, vom forma o nouă ecuaţie corespunzător liniei sale (a treia).

Avem

$$d_{14} = 1; \quad d_{15} = \frac{1/5}{1 - 1/5}(1 - 7/10) = 3/40; \quad r_{10} = 1/5.$$

Scanned with OKEN Scanner

Deci

$$x_6 - x_4 - \frac{3}{40} x_5 = -\frac{1}{5}$$

este o nouă relaţie adiţională.

Iniţiem un nou ciclu, completînd ultima parte a tabelului simplex 39, corespunzător ultimei relaţii, şi folosind algoritmul simplex dual pentru eliminarea componentelor negative din soluţie (tabelul 40).

Tabelul 40

C_B	c_j	→	1/4	1	0	0	0	0
↓	variabile bazice	P_0	P_1	P_2	P_3	P_4	P_5	P_6
1	x_2	1	0	1	0	0	1	0
0	x_3	3/20	0	0	1	−1/2	−17/20	0
1/4	x_1	6/5	1	0	0	1	− 3/10	0
0	←x_6	−1/5	0	0	0	$\boxed{-1}$	− 3/40	1
	$c_j - z_j$	$z_0 = 13/10$	0	0	0	−1/4	−37/40	0
1	x_2	1	0	1	0	0	1	0
0	x_3	1/4	0	0	1	0	13/16	−1/2
1/4	x_1	1	1	0	0	0	3/8	1
0	x_4	1/5	0	0	0	1	3/40	−1
	$c_j - z_j$	$z_0 = 5/4$	0	0	0	0	−35/32	−1/4

Noua soluţie obţinută,

$$x_1 = 1,\ x_2 = 1,\ x_3 = 1/4,\ x_4 = 1/5,$$

avînd primele două componente întregi, este soluţia programului mixt.

Scanned with OKEN Scanner

§ 4. PROGRAMAREA PĂTRATICĂ DISCRETĂ

Fie programul pătratic discret

(1) $\min\{f(X) = dX + X'CX | AX \leqslant b,\ X \geqslant 0,\ x_i$ întregi, $i = 1, 2, \ldots, n\}$

C, fiind o matrice nenegativ definită.

La baza metodei de rezolvare a acestei probleme, stă ideea de a înlocui programul (1) printr-o secvenţă finită de programe liniare mixte.

Vom descrie, în cele ce urmează, algoritmul şi vom da pe scurt şi justificarea sa.

Fie X^0, punctul de minim global al funcţiei convexe $f(X)$ în absenţa restricţiilor.

Într-o etapă preliminară se calculează, cu una din metodele programării pătratice, soluţia $\overline{X}$ a programului neîntreg

(2) $$\min\{f(X) = dX + X'CX | AX \leqslant b,\ X \geqslant 0\}.$$

Din definiţia lui X^0 rezultă că în acest punct trebuie să se anuleze toate derivatele lui f.

Cum

$$\frac{\partial f}{\partial x_j} = d_j + 2\sum_{i=1}^{n} c_{ji}x_i.$$

Rezultă

$$\sum_{i=1}^{n} c_{ji}x_i^0 = -d_j$$

sau, vectorial,

$$CX^0 = -\frac{1}{2}d',\ X^{0\prime}C = -\frac{1}{2}d$$

De aici

(3) $$f(X^0) = \frac{1}{2}dX^0.$$

Scanned with OKEN Scanner

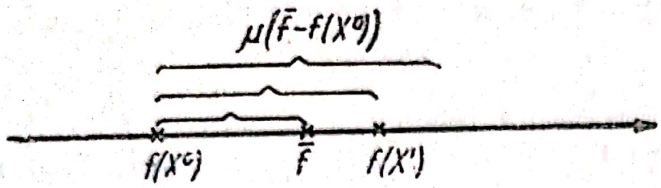


Fig. 18.

Să notăm

$$\bar{f} = f(\bar{X}).$$

Dacă soluţia $\bar{X}$ a programului (2) are toate componentele întregi, atunci ea constituie şi soluţia căutată a problemei (1).

În caz contrar se aplică prima iterată a algoritmului.

Ne propunem, pentru aceasta, să determinăm un vector X^1, soluţie posibilă întreagă a problemei (1), pentru care f să ia o valoare cît mai apropiată de cea optimă $\bar{f}$.

Deoarece X^0 este optimul fără restricţii, este clar că

$$f(X^0) \leqslant \bar{f} \leqslant f(X^1)$$

În ipotezele obişnuite de mărginire, putem afirma că inegalitatea:

$$f(X^1) - f(X^0) \leqslant \mu(\bar{f} - f(X^0)) \tag{4}$$

este verificată pentru o infinitate de valori reale.

Intuitiv, condiţia ca $f(X^1)$ să aibe o valoare cît mai apropiată de $\bar{f}$ (fig. 18) este echivalentă cu determinarea minimului valorilor reale μ ce satisfac inegalitatea (4).

Fie hipersuprafaţa

$$f(X) = \bar{f}$$

şi fie $\overline{X^1}$ un punct situat la intersecţia ei cu segmentul determinat de X^0 şi X^1.

Aceasta înseamnă că,

$$f(X^1) = \bar{f}, \tag{5}$$

$$\overline{X^1} = \lambda_1 X^1 + (1 - \lambda_1) X^0, \text{ cu } \lambda_1 \in [0, 1]. \tag{6}$$

Valoarea lui λ_1 rezultă din (5) şi (6) în urma unui calcul:

$$\bar{f} = f[\lambda_1 X^1 + (1 - \lambda_1) X^0] = \lambda_1 dX^1 + (1 - \lambda_1) dX^0 +$$
$$+ \lambda_1^2 (X^1)' C X^1 + \lambda_1 (1 - \lambda_1)(X^1)' C X^0 + \lambda_1 (1 - \lambda_1) X^0 C X^1 +$$
$$+ (1 - \lambda_1)^2 (X^0)' C X^0.$$

Scanned with OKEN Scanner

De aici şi din (3),

$$\bar{f} = \lambda_1^2\, dX^1 + \lambda_1^2 (X^1)'CX^1 + \frac{1}{2}(1 - \lambda_1^2) dX^0$$

adică,

$$\bar{f} - f(X^0) = \lambda_1^2 f(X^1) - \lambda_1^2 f(X_0).$$

Rezultă

$$\lambda_1 = \sqrt{\frac{\bar{f} - f(X^0)}{f(X^1) - f(X^0)}}. \tag{7}$$

Iată acum, descrierea primei iterate.
Se rezolvă programul liniar mixt.

$$AX \leqslant b,$$
$$X \geqslant 0$$
$$(d + \overline{X}'C)(X - X^0) \leqslant 2\,\mu\,[\bar{f} - f(X^0)], \tag{8}$$
$$x_i \text{ întregi, } i = 1, 2, \ldots, n;$$

minim μ.

Fie X^1 soluţia sa şi μ_1 minimul determinat.
Se calculează λ_1 din (7) şi $\overline{X}^1$ din (6).
Testăm optimalitatea lui X^1 astfel: dacă

$$[d + 2(\overline{X}^1)'C](X^1 + X^0) \leqslant 2\,\mu_1\,[\bar{f} - f(X^0)] \tag{9}$$

atunci, X^1 este soluţia optimă a programului (1).
Pentru justificare, scoatem din (6)

$$X^1 - X^0 = \frac{1}{\lambda_1}(X^1 - X^0)$$

Atunci din (9),

$$2\mu_1\,[\bar{f} - f(X^0)] \geqslant \frac{1}{\lambda_1}\,[d + 2(\overline{X}^1)'C]\cdot(\overline{X} - X^0) =$$

$$= \frac{1}{\lambda_1}\,[d\overline{X}^1 - dX^0 + 2(\overline{X}^1)'C\overline{X}^1 - 2(\overline{X}^1)'CX^0] =$$

$$= \frac{1}{\lambda_1}\,[d\overline{X}^1 - dX^0 + 2(\overline{X}^1)'C\overline{X}^1 + d\overline{X}^1] = \frac{2}{\lambda_1}\,(\bar{f} - f(X^0))$$

Scanned with OKEN Scanner

şi deci

$$\mu_1 \geqslant 1/\lambda_1.$$

Dar ţinînd seama de (7), rezultă de aici că μ_1 verifică (4).

Pe de altă parte, din (8) şi (9) va rezulta că μ_1 este cel mai mic, în ipoteza că X^1 este întreg.

Dacă testul (9) dă răspuns negativ, se iniţiază o nouă iterată, în care programului liniar mixt (8) i se adaugă o nouă restricţie de tipul (9).

Să presupunem că după un şir de k iterate, am obţinut vectorii întregi $X^1, X^2, \ldots, X^k$ şi vectorii aparţinînd hipersuprafeţei $f(X) = \bar{f}$, $\overline{X}^1, \overline{X}^2, \ldots, \overline{X}^k$

Procesul se încheie aici, dacă una din următoarele alternative se realizează:

a) $\overline{X}^k$ întreg (aceasta deoarece $f(\overline{X}^k) = \bar{f}$);

b) $[d + 2(\overline{X}^k)'C](X^k - X^\circ) \leqslant 2\mu_k (\bar{f} - f(X^\circ))$.

Dacă niciuna din alternative nu este îndeplinită, se începe o nouă iteraţie prin determinarea lui X^{k+1} ca soluţie a programului:

$$AX \leqslant b,$$

$$X \geqslant 0$$

$$(8') \quad (d + \overline{X}^j C)(X - X^\circ) \leqslant 2\mu \, [\bar{f} - f(X^\circ)], \quad j = 1, 2, \ldots, k.$$

$$x_i \text{ întregi}, \; i = 1, 2, \ldots, n$$

minim μ.

Putem nota, în încheiere, faptul că în anumite ipoteze, se poate demonstra convergenţa algoritmului către soluţia optimă întreagă, după un număr finit de paşi. Evident acest lucru se poate face ţinînd seama şi de ipotezele în care are loc convergenţa algoritmului pentru programul liniar discret mixt ce trebuie rezolvat în fiecare iterată.

Scanned with OKEN Scanner

Capitolul V

PROGRAMAREA LINIARĂ PARAMETRICĂ

§ 1. FORMULAREA PROBLEMEI

Dezvoltarea şi diversificarea modelelor programării liniare a creiat premiza studierii unor probleme care caută să răspundă cu o acurateţe sporită cerinţelor formulate de practică.

Modelele programării liniare prezentate în capitolul II, făceau posibilă determinarea politicilor optime ce trebuiau urmate în fenomenele modelate, cu condiţia ca toţi coeficienţii problemei să fie complet determinaţi.

De multe ori, problemele ridicate de practică conduc la considerarea, în modelul matematic corespunzător, a unor coeficienţi variabili. Dacă variaţia acestor coeficienţi are un caracter deterministic, atunci se poate urmări evoluţia soluţiei problemei, ca efect al acestei variaţii.

Programarea liniară parametrică îşi propune să studieze acele programe liniare în care toţi, sau o parte din coeficienţi pot fi consideraţi funcţii de unul sau mai mulţi parametri, cu condiţia ca mulţimile de valori ale acestor parametri să fie complet specificate. De fapt, scopul urmărit este acela de a preciza corecţiile care trebuie aplicate soluţiilor optime ale programului pe măsură ce parametri ce intervin aici variază, urmînd ca, apoi, planificatorul să adopte o decizie pe baza acestor rezultate.

Să urmărim în continuare un exemplu de problemă din practica economică al cărui model matematic este un program liniar parametric.

O uzină trebuie să satisfacă cererea faţă de un anumit produs, cerere care variază lunar pe parcursul unei perioade date.

Să presupunem că se cunoaşte cu precizie cantitatea de produs cerută de consumator în fiecare lună.

Scanned with OKEN Scanner

O variaţie mare, de la o lună la alta, a cererii, poate provoca dificultăţi considerabile producătorului. Într-adevăr, dacă şi-ar propune să producă în fiecare lună exact cantitatea cerută, uzina ar suferi pierderi datorate fie suplimentării producţiei, în lunile cu cerere mare, fie încetinirii artificiale a ritmurilor de producţie, în lunile cu cerere scăzută.

Acest inconvenient poate fi atenuat prin crearea unor stocuri lunare, care să acopere cererea sporită, din lunile de vîrf, pe seama supraproducţiei din lunile în care cererea este mică.

Trebuie ţinut seama, însă, de cheltuielile de stocaj care apar în cazul aplicării unei asemenea politici.

O decizie optimă poate fi luată numai în urma rezolvării unui program matematic care va încerca să indice un mod raţional de planificare a producţiei, astfel ca pierderile producătorului să fie minime.

Să introducem următoarele notaţii:

x_i — producţia planificată în luna i;

c_i — cererea de produs în luna i;

s_i — stocul pe care uzina îşi propune să-l creeze la sfîrşitul lunii i;

s_0 — stocul iniţial de care dispune uzina la începutul perioadei pentru care se face planificarea (poate fi 0);

a — cheltuielile suplimentare pentru unitatea de produs cauzate de creşterea producţiei de la o lună la alta;

b — cheltuielile de stocare lunare pentru unitatea de produs.

Deoarece cererea trebuie să fie întotdeauna satisfăcută integral, rezultă că la sfîrşitul fiecărei luni uzina trebuie să dispună de o cantitate de produs superioară cererii. Or, această cantitate este suma dintre cantitatea produsă în cursul lunii respective şi stocul rezultat la sfîrşitul lunii precedente, deci

$$x_i + s_{i-1} \geqslant c_i, \qquad i = 1, 2, \ldots, n$$

(am notat cu n numărul de luni din perioada pentru care se face planificarea).

O formă echivalentă a relaţiilor de mai sus este

$$(1) \qquad x_i + s_{i-1} - s_i = c_i, \qquad i = 1,2,\ldots,n,$$

$$s_i \geqslant 0, \qquad i = 0,1,\ldots,n.$$

Scanned with OKEN Scanner

Să punem acum,

$$x_i - x_{i-1} = u_i - v_i \tag{2}$$

cu $u_i \geqslant 0$, $v_i \geqslant 0$.

Evident, $u_i > 0$ şi $v_i = 0$ înseamnă o creştere a producţiei de la luna $i-1$ la luna i iar $u_i = 0$ şi $v_j > 0$ înseamnă o micşorare a producţiei în acelaşi interval.

Din prezentarea modelului, rezultă că uzina suportă cheltuieli suplimentare atunci cînd $u_i > 0$. Luînd în considerare şi cheltuielile de stocaj, ajungem la următoarea formulare a modelului matematic:

Să se determine valorile nenegative ale variabilelor x_i, s_i, u_i, şi v_i, care în condiţiile (1) şi (2) minimizează expresia cheltuielilor suplimentare.

$$a \sum_{i=1}^{n} u_i + b \sum_{i=1}^{n} s_i.$$

Dacă a şi b sînt constante, atunci problema enunţată aici devine un program liniar obişnuit. De cele mai multe ori, însă, aceşti coeficienţi nu pot fi precizaţi şi ca atare, planificatorul este interesat să cunoască comportarea soluţiei optime a problemei pentru diversele valori posibile ale lui a şi b.

Forma matematică generală a unei probleme de programare liniară parametrică este următoarea:

$$A(\lambda_1, \lambda_2, \ldots, \lambda_\gamma) \times \{\geqslant, =, \leqslant\} b(\lambda_1, \lambda_2, \ldots, \lambda_\gamma),$$
$$X \geqslant 0;$$

$$\max (\min)\, C(\lambda_1, \lambda_2, \ldots, \lambda_\gamma) X$$

unde componentele matricei A şi ale vectorilor b şi C sînt funcţii oarecare de parametrii $\lambda_1, \lambda_2, \ldots, \lambda_\gamma$ care au domenii de variaţie bine precizate.

Bine înţeles, rezolvarea problemei într-o formă atît de generală ridică dificultăţi substanţiale.

Rezultate bune s-au obţinut în cazuri particulare în care parametrii apar doar la o parte din coeficienţii problemei (numai în componentele vectorului b, sau numai în coeficienţii funcţiei obiectiv) şi în care aceşti coeficienţi sînt funcţii liniare de parametrii respectivi.

Scanned with OKEN Scanner

Vom lua în considerare, pentru început, problema de forma:

(1) $AX = b$,

(2) $X \geqslant 0$;

(3) $\max CX$

unde $C = C^0 + \lambda C^1$, cu C^0 și C^1 vectori linie m-dimensionali dați și λ un parametru real.

Să presupunem că pentru o valoare particulară λ_0 a parametrului λ, problema de programare liniară dedusă din (1), (2), (3), are optim finit și algoritmul simplex ne-a dat o soluție optimă X^0.

Aceasta înseamnă că

$$c_j - z_j \leqslant 0, \qquad j = 1,2,\ldots,n \tag{4}$$

unde
$z_j = C_B Z_j$, B fiind baza asociată lui X^0 (vezi notațiile din capitolul II)

Or, ținînd seama de forma particulară a lui C, rezultă

$$z_j = C_B^0 Z_j + \lambda_0 C_B^1 Z_j$$

și deci

$$c_j - z_j = (c_j - C_B^0 Z_j) - \lambda_0 C_B^1 Z_j.$$

Să notăm, pentru simplificare,

$$\alpha_j = c_j - C_B^0 Z_j,$$

$$\beta_j = - C_B^1 Z_j.$$

Atunci (4) devine.

$$\alpha_j + \lambda_0 \beta_j \leqslant 0, \qquad j = 1,2,\ldots,n. \tag{5}$$

O primă întrebare care se pune este următoarea: există și alte valori ale lui λ pentru care X^0 reprezintă soluție optimă a programului corespunzător?

Scanned with OKEN Scanner

Evident, pentru orice asemenea valoare a lui λ, condiţiile (5) ar trebui satisfăcute de λ, adică

$$\alpha_j + \beta_j \leqslant 0, \qquad j = 1,2,\ldots,n, \tag{6}$$

Este uşor de verificat, ţinînd seama de (5), că ultimele inegalităţi sînt satisfăcute pentru toate valorile lui λ cuprinse în intervalul $[\underline{\lambda}_0, \overline{\lambda}_0]$, unde

$$\underline{\lambda}_0 = \begin{cases} \max\limits_{\beta_j < 0} -\dfrac{\alpha_j}{\beta_j}, & \text{dacă există } j \text{ cu } \beta_j < 0, \\ -\infty, & \text{dacă } \beta_j \geqslant 0,\ j = 1, 2, \ldots, n; \end{cases} \tag{7}$$

$$\overline{\lambda}_0 = \begin{cases} \min\limits_{\beta_j > 0} -\dfrac{\alpha_j}{\beta_j}, & \text{dacă există } j \text{ cu } \beta_j > 0, \\ +\infty, & \text{dacă } \beta_j \leqslant 0,\ j = 1, 2, \ldots, n. \end{cases} \tag{8}$$

Avem deci

Teorema 1. *Dacă X^0 este soluţie optimă a programului liniar* (1), (2), (3), *corespunzător valorii λ_0 a parametrului, atunci va fi soluţie optimă pentru orice alt program liniar corespunzător unei valori din intervalul $[\underline{\lambda}_0, \overline{\lambda}_0]$.*

Odată determinat intervalul de variaţie al lui λ, pentru care X^0 este soluţie optimă, se pune în mod natural întrebarea dacă pentru alte valori ale parametrului, programul are încă soluţie optimă.

Răspunsul la această întrebare se poate da extinzînd treptat intervalul determinat anterior şi analizînd programele corespunzătoare noilor valori λ luate în considerare.

Să începem cu valorile $\lambda > \overline{\lambda}_0$.

Dacă $\overline{\lambda}_0 = +\infty$, discuţia este încheiată.

Să presupunem că $\overline{\lambda}_0$ este finit.

Aceasta înseamnă că există indici pentru care $\beta_j > 0$, şi fie k indicele pentru care este atins minimul rapoartelor $-\dfrac{\alpha_j}{\beta_j}$ cu $\beta_j > 0$.

Atunci, din (8)

$$\overline{\lambda}_0 = -\frac{\alpha_k}{\beta_k}.$$

Scanned with OKEN Scanner

Pentru orice $\lambda > \bar{\lambda}_0$, va rezulta

$$c_k - z_k = \alpha_k + \lambda\beta_k > \alpha_k - \frac{\alpha_k}{\beta_k}\beta_k = 0.$$

Două situaţii pot avea loc:

(i) toţi $z_{kj} \leqslant 0$, $j \in \bar{R}$ ($\bar{R}$ este mulţimea indicilor bazici);
(ii) există $j \in \bar{R}$, cu $z_{kj} > 0$.

În cazul (i) rezultă din teorema 3 cap. II. § 5, că programul va avea optim infinit pentru orice $\lambda > \bar{\lambda}_0$.

În cazul (ii) soluţia x^0, care a încetat de a mai fi optimă pentru $\lambda > \bar{\lambda}_0$, poate fi îmbunătăţită. Pentru aceasta vectorul P_k va înlocui un vector bazic P_j, pentru care $z_{kj} > 0$, conform algoritmului simplex uzual.

Va rezulta, ca efect, o nouă soluţie de bază X^1.

Se poate demonstra următorul rezultat:

Teorema 2. *X^1 este soluţia optimă pentru cel puţin un program corespunzător unei valori λ. Mulţimea tuturor valorilor λ pentru care X^1* reprezintă soluţia optimă a programului asociat este de forma $\underline{\lambda}_1 \leqslant \lambda \leqslant \bar{\lambda}_1$, *cu* $\underline{\lambda}_1 = \bar{\lambda}_0$.

Discuţia precedentă se repetă, dacă este cazul, pentru extinderea treptată a mulţimii de variaţie pînă la $+\infty$. Se ajunge astfel la o partiţionare a intervalului $[\bar{\lambda}_0, +\infty)$, în subintervale învecinate, fiecărui asemenea interval fiindu-i asociată o soluţie optimă. (Eventual, pentru ultimul interval, soluţia poate fi infinită).

Procesul se încheie după un număr finit de paşi, deoarece numărul bazelor extrase din matricea A este finit, iar fiecare etapă a procesului produce o nouă soluţie de bază.

O extindere similară se poate face pentru valorile $\lambda < \underline{\lambda}_0$.

Dacă $\underline{\lambda}_0 = -\infty$, procesul de extindere este încheiat.

Dacă $\underline{\lambda}_0$ este finit, atunci fie k indicele pentru care

$$-\frac{\alpha_k}{\beta_k} = \max_{\beta_j < 0} -\frac{\alpha_j}{\beta_j} = \underline{\lambda}_0,$$

Oricare ar fi $\lambda < \underline{\lambda}_0$, ţinînd seama că $\beta_k < 0$, avem

$$c_k - z_k = \alpha_k + \lambda\beta_k > \alpha_k - \frac{\alpha_k}{\beta_k}\beta_k = 0.$$

Scanned with OKEN Scanner

Aceleaşi două situaţii descrise anterior se pot produce.
În cazul (ii), algoritmul simplex dă o nouă soluţie de bază.

Teorema 3. *Noua soluţie este soluţie optimă pentru toate programele liniare corespunzătoare valorilor* $\lambda \in [\underline{\lambda}', \overline{\lambda}']$, *în plus,* $\overline{\lambda}' = \underline{\lambda}_0$.

Să presupunem acum, că pentru valoarea λ_0 iniţial acordată parametrului λ, algoritmul simplex ne indică nemărginirea optimului.

Va exista deci un vector P_j, în afara bazei iniţiale B, pentru care

$$c_j - z_j = \alpha_j + \lambda_0 \beta_j > 0, \tag{9}$$

$$z_{j_i} \leqslant 0, \; i \in \overline{R}. \tag{10}$$

Se pot ivi două cazuri:

(i') $\beta_j \geqslant 0$,
(ii') $\beta_j < 0$.

În cazul (i'), pentru orice $\lambda > \lambda_0$, inegalitatea (9) se păstrează şi atunci (10) ne arată că soluţia se va menţine infinită.

În cazul (ii'), inegalitatea (9) se păstrează pentru valorile de intervalul $[\lambda_0, \lambda']$ cu

$$\lambda' = -\frac{\alpha_j}{\beta_j}.$$

Ca atare programul admite optim infinit pentru toate valorile parametrului din intervalul determinat. Problema rămînînd deschisă pentru valorile superioare lui λ', se repetă analiza prezentată începînd cu $\lambda = \lambda'$.

În mod analog se procedează pentru $\lambda < \lambda_0$.

Dacă $\beta_j \leqslant 0$, pentru orice $\lambda < \lambda_0$ programul va avea optim infinit.

Dacă $\beta_j > 0$, aceasta va fi adevărat pentru $\lambda \in (\lambda'', \lambda_0]$, cu $\lambda'' = -\frac{\alpha_j}{\beta_j}$, de la acest punct discuţia reluîndu-se.

În concluzie, rezolvarea unui program liniar parametric, de tipul descris în prezentul paragraf, se reduce la determinarea unui număr finit de valori ale parametrului (numite valori caracteristice) care delimitează intervale pentru care soluţia se menţine

Scanned with OKEN Scanner

neschimbată. Metoda de rezolvare se bazează, în întregime pe algoritmul simplex uzual.

Dăm în continuare rezolvarea unui exemplu numeric.

Exemplu.

Fie programul parametric

$$\begin{aligned} x_1 - 2x_2 + 21x_3 - 2x_4 &= 0, \\ 2x_3 + x_4 + x_5 &= 8, \\ x_2 - 11x_3 + x_6 &= 10, \\ x_i \geqslant 0,\ i = 1,2,\ldots,6; & \end{aligned}$$

$$\max \quad (2-\lambda)x_1 + 2\lambda x_2 + (3-\lambda)x_3 + (-1+\lambda)x_4 + (5+10\lambda)x_5 + (1+4\lambda)x_6).$$

Căutăm soluţia programului pentru o valoare particulară a parametrului, de exemplu $\lambda = 0$ (tabelul 41).

Algoritmul simplex ne-a condus la soluţia optimă

$$x_1^0 = 20,\ x_2^0 = 10,\ x_5^0 = 8,\ x_3^0 = x_4^0 = x_6^0 = 0,$$

Conform teoremei 1 această soluţie este optimă pentru toate programele corespunzătoare valorilor λ cuprinse în intervalul $[\underline{\lambda}_0, \overline{\lambda}_0]$.

Determinăm valorile caracteristice $\overline{\lambda}_0$, $\underline{\lambda}_0$, conform relaţiilor (7) şi (8).

$$\underline{\lambda}_0 = -2/11,\ \overline{\lambda}_0 = 3/4.$$

Urmărim comportarea programului dat pentru valorile parametrului superioare lui 3/4.

Introducem în bază vectorul P_6 pentru care $\beta_6 = 4 > 0$ (tabelul 42).

Prima parte a tabelului simplex 42 conţine soluţia

$$x_2^1 = 0,\ x_5^1 = 8,\ x_6^1 = 10,\ x_1^1 = x_3^1 = x_4^1 = 0.$$

În baza teoremei 2, aceasta este soluţie optimă pentru toate programele corespunzînd valorilor parametrului din $[\underline{\lambda}_1, \overline{\lambda}_1]$.

Scanned with OKEN Scanner

Tabelul 41

$C_B = C_B^0 + \lambda C_B^0$	C_B^0	$c_j^0 + \lambda c_j^1$	→	$2 - \lambda$	2λ	$3 - \lambda$	$-1 + \lambda$	$5+10\lambda$	$1 + 4\lambda$
	↓	c_j^1	→	2	0	3	−1	5	1
↓		variabile bazice	P_0	P_1	P_2	P_3	P_4	P_5	P_6
$2 - \lambda$	2	x_1	6	1	−2	21	−2	0	0
$5 + 10\lambda$	5	x_5	8	0	0	2	1	1	0
$1 + 4\lambda$	←1	x_6	10	0	[1]	−11	0	0	1
		$c_j^0 - C_B^0 Z_j$		0	3	−38	−2	0	0
$2 - \lambda$	2	x_1	20	1	0	− 1	−2	0	[2]
$5 + 10\lambda$	5	x_5	8	0	0	2	①	1	0
2λ	0	x_2	10	0	1	−11	0	0	1
		$c_j^0 - C_B^0 Z_j$		0	0	−5	−2	0	−3
		$\alpha_j + \lambda\beta_j$		0	0	−5	$-2-11\lambda$	0	$-3+4\lambda$

Scanned with OKEN Scanner

Tabelul 42

$C_B = C_B^0 + \lambda C_B^1$	$c_B^0 + \lambda c_j^1$	$\rightarrow$	$2 - \lambda$	2λ.	$3 - \lambda$	$-1 + \lambda$	$5 + 10\lambda$	$1 + 4\lambda$
	variabile bazice	P_0	P_1	P_2	P_3	P_4	P_5	P_6
$1 + 4\lambda$	x_6	10	1/2	0	−1/2	−1	0	1
$5 + 10\lambda$	x_5	8	0	0	[2]	1	1	0
2λ	x_2	0	−1/2	1	−21/2	1	0	0
	$\alpha_j + \lambda\beta_j$		$3/2 - 2\lambda$	0	$-13/2+2\lambda$	$-5 - 7\lambda$	0	0
$1 + 4\lambda$	x_6	24	1/2	0	0	−3/4	1/4	1
$3 - \lambda$	x_3	4	0	0	1	1/2	1/2	0
2λ	x_2	84	−1/2	1	0	25/4	21/4	0
	$\alpha_j + \lambda\beta_j$		$3/2 - 2\lambda$	0	0	$-7/4-14\lambda$	$13/2-\lambda$	0

Scanned with OKEN Scanner

Valoarea caracteristică λ_1 este egală cu $\overline{\lambda}_0 = 3/4$, iar $\overline{\lambda}_1$ se determină conform (8); $\overline{\lambda}_1 = 13/9$.

În continuare, soluția poate fi mofidicată, în scopul optimizării programelor care corespund valorilor lui λ superioare lui 13/9.

Pentru aceasta se introduce în bază vectorul P_3 pentru care $\beta_3 = 2 > 0$.

A doua parte a tabelului simplex 42 conține elementele corespunzătoare transformării de bază efectuată și dă soluția

$$x_2^2 = 84,\ x_3^2 = 4,\ x_6^2 = 24,\ x_1^2 = x_4^2 = x_5^2 = 0.$$

Deoarece toți $\beta_j < 0$, această soluție este optimă pentru toți $\lambda \in [13/9, +\infty)$.

Pentru analiza valorilor lui λ mai mici decît $-2/11$, plecînd de la tabloul simplex corespunzător soluției X^0, introducem în bază vectorul P_4 pentru care $\beta_4 < 0$ (tabelul 43).

Tabelul 43

$C_B = C_B^0 + \lambda C_B^1$	$c_j^0 + c_j^1$	→	$2-\lambda$	2λ	$3-\lambda$	$-1+\lambda$	$5+10\lambda$	$1+4\lambda$
	variabile bazice	P_0	P_1	P_2	P_3	P_4	P_5	P_6
$2-\lambda$	x_1	28	1	0	3	0	2	2
$-1+\lambda$	x_4	8	0	0	2	1	1	0
2λ	x_2	10	0	1	-11	0	0	1
	$\alpha_j + \lambda\beta_j$		0	0	$-1+22\lambda$	0	$2+11\lambda$	$-3+4\lambda$

Noua soluție,

$$x_1' = 28,\ x_2' = 10,\ x_4' = 8,\ x_3' = x_5' = x_6' = 0$$

este optimă pentru toți $\lambda < -2/11$, deoarece $\beta_k > 0$, $k = 1,2,\ldots,6$.

În concluzie, programul parametric dat admite soluții optime pentru toate valorile reale ale parametrului λ. Prin aplicarea

Scanned with OKEN Scanner

tehnicii de calcul descrisă în acest paragraf, am obţinut o divizare a mulţimii numerelor reale în patru intervale, prin valorile caracteristice $\underline{\lambda}_0 = -2/11$, $\bar{\lambda}_0 = 3/4$, $\bar{\lambda}_1 = 11/9$. Corespunzător intervalelor $(-\infty, -2/11]$, $[-2/11, 3/4]$, $[3/4, 11/9]$, $[11/9, +\infty)$, soluţiile optime sînt X' X^0, X^1, X^2, obţinute anterior.

§ 3. PARAMETRIZAREA VECTORULUI b

Fie programul liniar

(1) $AX = b$,

(2) $X \geqslant 0$,

(3) $\max CX$

unde componentele vectorului b sînt funcţii liniare de un parametru real λ. Vom scrie, atunci

$$b = b^0 + \lambda b^1.$$

Putem presupune că $b^0 \geqslant 0$.

Ca şi în cazul discutat în §2 este interesant de urmărit variaţia soluţiei optime a programelor de forma (1), (2), (3), corespunzătoare diferitelor valori posibile ale parametrului λ.

Fie λ_0 o valoare oarecare din domeniul de variaţie al lui λ.

Să presupunem că algoritmul simplex aplicat programului liniar obişnuit, obţinut din (1), (2), (3), prin înlocuirea lui λ cu λ_0, ne-a condus la o soluţie optimă X^0.

Avem deci

(4) $c_j - z_j \leqslant 0, \qquad j = 1,2,\ldots,n.$

Dacă B este baza corespunzătoare lui X°, atunci, componentele bazice ale soluţiei optime X° sînt componentele vectorului

(5) $B^{-1}b = B^{-1}b^0 + \lambda_0 B^{-1} b^1$

Conform (5), vom scrie fiecare componentă bazică a soluţiei optime X^0 sub forma

(6) $x_i^0 = \alpha_i + \lambda\beta_i, \qquad i \in \bar{R}.$

Scanned with OKEN Scanner

Vom răspunde întîi la întrebarea: pentru ce valori ale lui λ, X^0 rămîne încă soluţie optimă?

Este clar că atîta timp cît baza B rămîne neschimbată, testul de optimalitate (4) este verificat. Drept urmare X^0 va constitui soluţie optimă pentru toate programele corespunzătoare acelor valori ale lui λ pentru care componentele bazice ale soluţiei rămîne nenegative.

Putem formula, deci astfel, primul rezultat.

Teorema 1. *X^0 este soluţia optimă pentru toate programele deduse din* (1), (2), (3) *corespunzător valorilor λ satisfăcînd*

$$(7)\quad \underline{\lambda}_0 \leqslant \lambda \leqslant \overline{\lambda}_0$$

unde

$$(8)\quad \underline{\lambda}_0 = \begin{cases} \max\limits_{\beta_j < 0} -\dfrac{\alpha_j}{\beta_j}, & \text{dacă există } j \text{ cu } \beta_j > 0, \\ -\infty, & \text{dacă } \beta_j \leqslant 0,\ j = 1,2,\ldots,n; \end{cases}$$

$$(9)\quad \overline{\lambda}_0 = \begin{cases} \min\limits_{\beta_j < 0} -\dfrac{\alpha_j}{\beta_j}, & \text{dacă există } j \text{ cu } \beta_j < 0, \\ +\infty, & \text{dacă } \beta_j \geqslant 0,\ j = 1,2,\ldots,n. \end{cases}$$

Ne interesează, în continuare, posibilităţile de extindere a mulţimii valorilor lui λ pentru care programul admite soluţii optime, precum şi modul de obţinere a acestora din soluţia iniţială.

Dacă $\overline{\lambda}_0 = +\infty$, evident, extinderea spre dreapta a mulţimii anterioare este încheiată.

Să presupunem că $\overline{\lambda}_0$ este finit.

Fie atunci k, pentru care

$$-\frac{\alpha_k}{\beta_k} = \min_{\beta_j < 0} -\frac{\alpha_j}{\beta_j} = \overline{\lambda}_0.$$

Pentru orice $\lambda > \overline{\lambda}_0$, componenta bazică corespunzătoare a lui X^0 va satisface relaţia

$$x_k^0 = \alpha_k + \lambda\beta_k < \alpha_k + \overline{\lambda}_0\beta_k = \alpha_k - \frac{\alpha_k}{\beta_k}\beta_k = 0.$$

Scanned with OKEN Scanner

Deci, X^0 nu va mai fi soluţie posibilă, dar datorită relaţiilor (4) putem afirma că va fi o soluţie dual-posibilă.

Se poate încerca, prin urmare, folosind algoritmul simplex dual, să se modifice convenabil X^0, astfel încît noua soluţie rezultată să fie una optimă.

Două situaţii pot avea loc:

(i) toţi $z_{ik} \geqslant 0$,

(ii) există $i \in R$ astfel ca $z_{ik} < 0$.

În primul caz programul nu va admite soluţii posibile şi deci nici soluţii optime, pentru nici o valoare $\lambda > \bar{\lambda}_0$.

În cazul (ii) se modifică baza B prin înlocuirea vectorului P_k cu un vector P_i. Fie X^1 soluţia dual-posibilă corespunzătoare noii baze. Se poate demonstra următorul rezultat:

Teorema 2. *X^1 este soluţie optimă pentru toate programele corespunzătoare valorilor $\lambda \in [\underline{\lambda}_1 \ \bar{\lambda}_1]$; în plus, $\underline{\lambda}_1 = \bar{\lambda}_0$*

Continuînd în acest mod procesul, vom epuiza, după un număr finit de paşi intervalul $[\bar{\lambda}_0, +\infty)$, obţinînd o suită de soluţii optime, valabile fiecare pentru cîte un subinterval de valori ale parametrului λ, sau, constatăm imposibilitatea rezolvării programului dincolo de o anumită valoare a lui λ. Toate soluţiile optime vor fi obţinute cu algoritmul simplex dual.

Într-un mod asemănător se va face discuţia valorilor parametrului situate la stînga lui λ_0.

Dacă $\underline{\lambda}_0 = -\infty$, orice discuţie este, evident, de prisos.

Să presupunem că $\underline{\lambda}_0$ este finit.

Fie k indicile pentru care

$$-\frac{\alpha_k}{\beta_k} = \max_{\beta_j > 0} -\frac{\alpha_j}{\beta_j} = \underline{\lambda}_0.$$

Cum pentru orice $\lambda < \underline{\lambda}_0$, va rezulta, în virtutea relaţiei (8),

$$x_k^0 < 0$$

se pot lua în considerare aceleaşi două cazuri (i) şi (ii) analizate în discuţia precedentă.

În ipoteza că situaţia (ii) a fost adevărată, pentru noua soluţie dual-posibilă dată de algoritmul simplex dual, se poate formula următoarea

Teorema 3. *Noua soluţie este optimă pentru toate programele corespunzătoare valorilor $\lambda \in [\underline{\lambda}_1', \bar{\lambda}']$; iar $\bar{\lambda}_1' = \underline{\lambda}_0$.*

Scanned with OKEN Scanner

Să analizăm acum situaţia ce se poate creea dacă pentru valoarea iniţială λ_0 acordată parametrului λ, programul corespunzător nu are soluţie.

Se poate determina atunci o bază B, astfel ca X^0 să fie o soluţie dual-posibilă. În acelaşi timp, deoarece programul nu are soluţii, va exista o componentă bazică $x_i^0 < 0$, $i \in \bar{R}$ şi $z_{ij} \geqslant 0$, pentru $j = 2, \ldots, n$.

Ne putem afla în una din următoarele două situaţii:

(i') $\beta_i \leqslant 0$,

(ii') $\beta_i > 0$.

În cazul (i'), pentru orice $\lambda > \lambda_0$,

$$\alpha_i + \lambda\beta_i \leqslant \alpha_i + \lambda_0\beta_i < 0$$

şi deci, pentru nici o valoare a parametrului superioară lui λ_0 programul nu va avea soluţii.

În cazul (ii') putem afirma că programul nu are soluţii pentru valorile $\lambda \in [\lambda_0, \lambda')$, unde $\lambda' = -\frac{\alpha_i}{\beta_i}$.

În continuare se caută să se rezolve programul pentru valoarea λ' a parametrului şi se reîncepe de aici încolo o nouă discuţie pe baza rezultatului obţinut.

Pentru valorile parametrului inferioare lui λ_0 se foloseşte un raţionament asemănător, ajungînd la următoarele concluzii:

— dacă $\beta_i \geqslant 0$, programul nu are soluţii pentru nici o valoare $\lambda < \lambda_0$.

— dacă $\beta_i < 0$, programul nu are soluţii pentru $\lambda \in (\lambda', \lambda_0]$, discuţia reîncepînd de la valoarea $\lambda' = -\frac{\alpha_i}{\beta_i}$.

Exemplu numeric.

Fie programul parametric

$$\begin{aligned} x_1 - 2x_2 + x_3 - 2x_4 &= 6 - \lambda, \\ 2x_3 + x_4 + x_5 &= 8 + 12\lambda, \\ 3x_2 - x_3 + x_6 &= 10 - 2\lambda, \\ x_i \geqslant 0,\ i &= 1,2,\ldots,6; \\ \max\ (2x_1 - 3x_2 + x_4 + 5x_5 + x_6). & \end{aligned}$$

Căutăm să rezolvăm programul pentru o valoare particulară a parametrului, de exemplu $\lambda_0 = 0$ (tabelul 44).

Scanned with OKEN Scanner

Tabelul 44

C_B ↓	c_j variabile bazice	$b_j^0 + \lambda b_j^1 = \alpha_j + \lambda\beta_j$	b_j^0 P_0	2 P_1	−3 P_2	0 P_3	1 P_4	5 P_5	1 P_6
2	x_1	$6 - \lambda$	6	1	−2	1	−2	0	0
5	x_5	$8 + 12\lambda$	8	0	0	2	1	1	0
1	x_6	$10 - 2\lambda$	10	0	3	$\boxed{-1}$	0	0	1
	$c_j - z_j$			0	−2	−11	0	0	0

Soluția de forma

$$x_1^0 = 6 - \lambda,\ x_5^0 = 8 + 12\lambda,\ x_6^0 = 10 - 2\lambda,\ x_2^0 = x_3^0 = x_4^0 = 0$$

este optimă pentru toate programele corespunzînd valorilor $\lambda \in [\underline{\lambda}_0, \overline{\lambda}_0]$ unde, din (8) și (9)

$$\underline{\lambda}_0 = -2/3,\ \overline{\lambda}_0 = 5.$$

Pentru analiza valorilor $\lambda > 5$, eliminăm din bază, folosind algoritmul simplex dual, vectorul P_6 care corespunde raportului $5 = 10/2 = \min\limits_{\beta_j < 0} -\frac{\alpha_j}{\beta_j}$ (tabelul 45).

Tabelul 45

C_B ↓	c_j variabile bazice	$b_j^0 + \lambda b_j^1 = \alpha_j + \lambda\beta_j$	2 P_1	−3 P_2	0 P_3	1 P_4	5 P_5	1 P_6
2	x_1	$16 - 3\lambda$	1	1	0	−2	0	1
5	x_5	$28 + 8\lambda$	0	6	0	1	1	2
0	x_3	$-10 + 2\lambda$	0	−3	1	0	0	−1
	$c_j - z_j$		0	−35	0	0	0	−11

Scanned with OKEN Scanner

Noul vector X^1 de componente

$$x_1^1 = 16 - 3\lambda, \quad x_3^1 = -10 + 2\lambda, \quad x_5^1 = 28 + 8\lambda, \quad x_2^1 = x_4^1 = x_6^1 = 0$$

va da soluţia optimă pentru toate programele corespunzătoare valorilor $\lambda \in [\lambda_1, \bar{\lambda}_1]$, cu $\lambda_1 = \bar{\lambda}_0 = 5$ şi $\bar{\lambda}_1 = 16/3$.

Mai departe încercăm să extindem ultimul interval căpătat, eliminînd din bază vectorul P_1 (tabelul 46).

Tabelul 46

C_B ↓	→ variabile bazice	$b_j^0 + \lambda b_j^1 = \alpha_j + \lambda\beta_j$	2	−3	0	1	5	1
			P_1	P_2	P_3	P_4	P_5	P_6
1	x_4	$-8 + \frac{3}{2}\lambda$	−1/2	−2	0	1	0	−1/2
5	x_5	$36 + \frac{13}{2}\lambda$	1/2	13/2	0	0	1	5/2
0	x_3	$-10 + 2\lambda$	0	0	1	0	0	−1
	$c_j - z_j$		0	−67/2	0	0	0	−11

Vectorul X^2 de componente

$$x_3^2 = -10 + 2\lambda, \quad x_4^2 = -8 + 3/2\lambda, \quad x_5^2 = 36 + 13/2\lambda,$$
$$x_1^2 = x_5^2 = x_6^2 = 0$$

reprezintă soluţia optimă pentru toate valorile $\lambda \in [16/3, +\infty)$ (deoarece toţi $\beta_j \geq 0$).

Pentru valorile $\lambda < -2/3$, examinînd primul tablou simplex, putem conclude că programul nu are soluţii. Intr-adevăr pentru indicele $k = 5$ pentru care $\beta_k > 0$, toţi $z_{ik} \geq 0$, ceea ce plasează discuţia în cazul (ii).

Scanned with OKEN Scanner

Capitolul VI

PROGRAMAREA LINIARĂ STOCHASTICĂ

§ 1. CONSIDERENTE GENERALE

1.1. Punerea problemei

O problemă de programare matematică este o problemă de decizie, scopul urmărit fiind determinarea acelor valori ale variabilelor care optimizează o anume funcţie obiectiv.

Valorile optime ale variabilelor depind însă de o serie de parametri care apar în programul dat (coeficienţii funcţiei obiectiv, resurse, coeficienţi tehnologici).

Este normal deci ca, atunci cînd o parte din aceşti parametri se schimbă, soluţia optimă să se modifice. Tehnica de calcul obişnuită pentru rezolvarea acestor probleme, bazată pe algoritmul simplex, dă soluţii optime numai în cazul în care sînt complet specificate mulţimile de valori ale parametrilor (cap. V).

Chiar şi aşa, trebuie remarcat faptul că prin asemenea metode nu se poate obţine expresia explicită a soluţiei ca funcţie de parametri variabili.

În multe cazuri practice, de mare importanţă, este necesar ca parametri care apar în problemă să fie priviţi ca variabile aleatoare. Acest fapt are implicaţii importante asupra aspectelor teoretice ale programării matematice, cît şi, mai ales, asupra metodelor de calcul, complicînd substanţial problema.

Ce se înţelege printr-un program liniar stochastic?

Dacă în problema

$$\max \text{ (min) } f(X),$$

$$AX \{ \geqslant, =, \leqslant \} b,$$

$$X \geqslant 0$$

matricea A precum şi vectorii b, C, au toate componentele numere date, atunci avem de a face cu un program liniar ordinar. Dacă însă, o parte din elementele menţionate sînt variabile

aleatoare cu o repartiţie probabilistică cunoscută, vom spune că s-a definit un program liniar stochastic.

Evident, nu lipsit de interes este şi cazul cînd repartiţiile mărimilor aleatoare ce apar în problemă nu sînt cunoscute, dar acest tip de probleme nu va fi abordat în prezentul capitol.

În majoritatea situaţiilor practice de mare interes, metodele de investigaţie statistică permit determinarea repartiţiilor menţionate, sau, măcar a unor caracteristici esenţiale ale acestora.

Pentru formalizarea problemei stochastice, fie $\{\Omega, K, P\}$, cîmpul de probabilitate pe care sînt definite variabilele aleatoare ce apar în definiţia programului stochastic.

Pentru fiecare $\omega \in \Omega$, adică pentru fiecare realizare a întîmplării, $A(\omega)$, $b(\omega)$, $C(\omega)$ au elemente bine precizate şi ca atare, rezultă o problemă de programare liniară deterministică pentru care se poate deduce, cu metodele obişnuite, soluţia optimă $X^*(\omega)$ şi valoarea optimă $z^*(\omega)$ a funcţiei f.

Dar o asemenea tratare „a posteriori" a problemei, care implică o tactică pasivă de aşteptare a realizării parametrilor, nu este cea mai indicată pentru o problemă de decizie.

O interpretare „a priori" a datelor problemei, conduce la concluzia că X^* şi z^* sînt variabile aleatoare, avînd repartiţii induse de repartiţiile parametrilor aleatori $A(\omega)$, $b(\omega)$, $C(\omega)$.

În consecinţă, un obiectiv important ce se pune în legătură cu un program stochastic constă în stabilirea legilor probabilistice ale variabilelor aleatoare X^*, z^*.

Determinarea repartiţiilor acestor variabile reprezintă o sarcină dificilă, chiar în cazul cînd se poate da o expresie explicită a lui X^* şi z^* în funcţie de parametrii A, b, C. Pentru multe probleme de programare acest lucru devine imposibil.

În următoarele două paragrafe ale acestui capitol, vom încerca să rezolvăm asemenea probleme — numite probleme de distribuţie — pentru două tipuri particulare de programe stochastice, folosind metoda descrierii complete.

Din punct de vedere practic, a da răspuns la o problemă de distribuţie, cînd acest lucru se poate, nu înseamnă o rezolvare completă a problemei stochastice.

Într-adevăr, determinarea „a priori" a legii probabilistice a soluţiei optime X^*, sau a valorii optime z^*, nu înseamnă decît cunoaşterea mulţimilor de valori ale acestora şi lasă sub semnul incertitudinii alegerea deciziei care trebuie să fie luată.

Scanned with OKEN Scanner

De aceea rezolvarea completă a unui program stochastic, implică combinarea problemelor de distribuţie cu o serie de alte probleme, în general de factură aleatoare, integrîndu-le unui proces complex de decizie.

De notat, că de multe ori în situaţiile practice care impun programe stochastice, acestea se înlocuiesc cu probleme deterministe în care parametri aleatori se înlocuesc cu valorile lor medii ţinînd seama, eventual, şi de dispersiile acestora. O asemenea manieră de reformulare a problemei am întîlnit-o în capitolul III (§ 1 şi § 3).

Rezultatele bune care se obţin astfel, în anumite cazuri particulare, nu micşorează însă interesul major de care se bucură problemele de programare stochastică abordate în întreaga lor complexitate.

1.2. Un exemplu

O fermă agricolă de stat dispune de 1 000 ha teren arabil şi de un fond de 2 400 000 lei. Ea poate însămînţa trei culturi diferite (A,B,C) în următoarele condiţii:

a) — cultura A produce în medie 1 800 kg la ha,
— cultura B produce în medie 2 400 kg la ha,
— cultura C produce în medie 2 000 kg la ha;

b) — cheltuielile necesitate de cultura A sînt 2 000 lei/ha,
— cheltuielile necesitate de cultura B sînt de 3 000 lei/ha,
— cheltuielile necesitate de cultura C sînt de 2 500 lei/ha;

c) — preţul de vînzare pentru cultura A este de 2 lei/kg,
— preţul de vînzare pentru cultura B este 2,58 lei/kg,
— preţul de vînzare pentru cultura C este de 2,50 lei/kg.

Pentru a-şi ridica, la nivel mai înalt, rentabilitatea, această fermă şi-a propus să stabilească de aşa manieră repartiţia celor trei culturi pe suprafaţa disponibilă, încît să realizeze un venit maxim posibil, dat fiind că nu există niciun alt considerent de ordin economic.

Fie dar, x_1 numărul de hectare cultivate cu cultura A, x_2 numărul de hectare afectate culturii B şi x_3 numărul de hectare afectate lui C.

Următoarele condiţii vor trebui satisfăcute cu necesitate:

Scanned with OKEN Scanner

$$x_1 + x_2 + x_3 = 1\,000,$$

(1) $2\,000\ x_1 + 3\,000\ x_2 + 2\,500\ x_3 \leqslant 2\,400\,000,$

$$x_1 \geqslant 0,\ x_2 \geqslant 0,\ x_3 \geqslant 0.$$

Beneficiul total va fi dat de expresia liniară.

(2) $2 \cdot 1\,800\ x_1 + 2{,}58 \cdot 2\,400\ x_2 + 2{,}50 \cdot 2\,000\ x_3 -$

$$-2\,000\ x_1 - 3\,000\ x_2 - 2\,500\ x_3 = 1\,600\ x_1 + 3\,200\ x_2 + 2\,500\ x_3.$$

Se remarcă faptul că aici calcularea beneficiului s-a făcut pe baza producţiilor medii ale celor trei culturi.

Se poate pune atunci o problemă de programare liniară care constă în a găsi maximul beneficiului calculat mai sus; deci trebuie ca în condiţiile (1) şi (2) să avem

(3) $1\,600\ x_1 + 3\,200\ x_2 + 2\,500\ x_3 = \text{maxim.}$

Cu ajutorul primei egalităţi din (1) se poate reduce numărul necunoscutelor la două punînd:

$$x_3 = 1\,000 - x_1 - x_2.$$

Atunci (1) (3) devin

$$x_1 - x_2 \geqslant 200,$$

$$x_1 + x_2 < 1\,000,$$

(4)

$$x_1 \geqslant 0,$$

$$x_2 \geqslant 0;$$

(5) $$-9x_1 + 7x_2 = \text{maxim.}$$

Grafic, soluţia problemei rezultă imediat (fig. 19).

Mulţimea soluţiilor posibile este triunghiul MNP, soluţia optimă găsindu-se într-unul din vîrfuri şi anume în M pentru care

$$x_1 = 200,\ x_2 = 0,\qquad x_3 = 0.$$

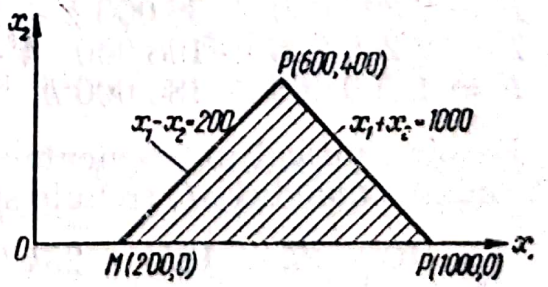


Fig. 19

Scanned with OKEN Scanner

Soluţia problemei nu poate fi considerată satisfăcătoare din punct de vedere economic. La întocmirea planului de cultură cunoaştem producţia medie la hectar, însă nu sîntem siguri că în realitate, după strîngerea recoltei, vom obţine tocmai această medie. Singura apreciere pe care o putem face este că producţia recoltată va fi, cu o anumită probabilitate, cuprinsă într-un anumit interval.

Vom presupune că pentru cele trei culturi, producţia la hectar se distribuie după legea normală cu următoarele valori tipice:

cultura	producţia medie	dispersia
A	1 600	180
B	3 200	360
C	2 500	250

În acest caz se poate afirma că producţia reală a culturii A va fi $1\,600 + 180\,k$, unde cu o probabilitate foarte mare (0,997) k este un număr cuprins între -3 şi $+3$.

Pentru celelalte culturi producţiile vor fi $3\,200 + 360\,k'$, respectiv, $2\,500 + 250\,k''$, k' şi k'' avînd aceeaşi semnificaţie ca mai sus.

Atunci beneficiul care trebuie să fie maximizat va avea următoarea expresie

$$(6)\quad F(x_1, x_2, x_3) = (1\,600 + 180\,k)x_1 + (3\,200 + 360\,k')x_2 + (2\,500 + 250\,k'')x_3.$$

Cum condiţiile (1) şi (3), respectiv (4), nu sînt afectate de reconsiderarea făcută asupra beneficiilor, maximul lui F va fi atins tot în unul din vîrfurile triunghiului MNP.

Calculînd valorile funcţiei F, în fiecare din punctele M, N, P, obţinem următoarele valori:

$F = 2\,320\,000 + 36\,000\,k + 200\,000\,k''$, în punctul M,
$F = 2\,240\,000 + 108\,000\,k + 144\,000\,k'$, în punctul N,
$F = 1\,600\,000 + 180\,000\,k$, în punctul P.

De aici, un calcul elementar ne arată că pentru ca maximul să cadă în punctul M, trebuie să fie îndeplinite condiţiile:

$$(7)\quad \begin{aligned} &10 - 9k - 18k' + 25k'' > 0, \\ &90 - 18k + 25k'' > 0. \end{aligned}$$

Scanned with OKEN Scanner

Maximul cade în P, dacă sînt îndeplinite condiţiile:

$$\begin{cases} 10 - 9k - 18k' + 25k'' < 0, \\ 80 - 9k + 18k' > 0. \end{cases} \tag{8}$$

În sfîrşit, poziţia de maxim corespunde punctului N, dacă:

$$\begin{cases} 90 - 18k + 25k'' < 0, \\ 80 - 9k + 18k' < 0, \end{cases} \tag{9}$$

Numerele k,k',k'', urmează fiecare o lege normală cu media 0 şi dispersia 1. Dacă presupunem că sînt independente, un calcul simplu de probabilităţi ne arată că:

(10) Prob. $(10 - 9k - 18k' + 25k'' > 0) = 0{,}5120,$

(10′) Prob. $(90 - 18k + 25k'' > 0) = 0{,}9987,$

(10″) Prob. $(80 - 9k + 18k' > 0) = 0{,}9999.$

De aici rezultă că inegalităţile (9) sînt îndeplinite cu o probabilitate mai mică decît 10^{-6}. Din punct de vedere practic, putem înlătura posibilitatea ca punctul N să fie soluţie optimă.

Ca atare, putem enunţa următorul rezultat:

— cu o probabilitate aproximativ egală cu 0,51 soluţia optimă este $x_1 = 200,\ x_2 = 0,\ x_3 = 800.$

— cu o probabilitate aproximativ egală cu 0,49 soluţia optimă este $x_1 = 600,\ x_2 = 400,\ x_3 = 0.$

Pentru luarea unei decizii planificatorul va trebui, oricum, să ţină cont de rezultatele obţinute anterior.

Totodată, este clar că acestea nu sînt suficiente pentru ca decizia să se bazeze exclusiv pe ele.

De exemplu, faptul că punctele M şi P au şanse aproximativ egale să devină optime, nu este de natură să creeze o anumită politică preferenţială.

De aceea, adiacent problemei de distribuţie care s-a rezolvat pînă acum, planificatorul trebuie să impună şi alte criterii, ţinînd de natura concretă economică.

Am putea cere, de exemplu, să se asigure un beneficiu care să nu fie sub un anumit nivel. Atunci este normal să alegem drept

Scanned with OKEN Scanner

politică optimă, punctul care asociază, cu o probabilitate maximă un cîştig superior pragului fixat.

Fixînd pragul inferior al cîştigului la 2 200 000, să vedem care din punctele M şi P ne garantează un cîştig superior acestui prag cu o probabilitate mai mare.

Ţinînd seama de (6), probabilitatea cu care beneficiul corespunzător punctului M depăşeşte pragul fixat va fi:

Prob. $(F \geqslant 2\,200\,000) = \text{Prob.}\ (36\,000k' + 200\,000\,k'' \geqslant -120\,000) = 1/2 + \Phi(0{,}6) = 0{,}73.$

Analog, probabilitatea cu care beneficiul corespunzător punctului P depăşeşte pragul fixat va fi:

Prob. $(F \geqslant 2\,200\,000) = \text{Prob.}\ (108\,000k + 144\,000\,k'' \geqslant -40\,000) = 1/2 + \Phi\,(0{,}22) = 0{,}58.$

(Am folosit aici, un binecunoscut rezultat al statisticii matematice potrivit căruia în ipoteza independenţei, o combinaţie liniară a unor variabile aleatoare normal distribuite, urmează tot o repartiţie normală. Φ este funcţia lui Laplace).

§ 2. PROGRAMAREA LINIARĂ STOCHASTICĂ CU VECTORUL C ALEATOR

Să ne oprim asupra unui program ale cărui restricţii liniare sînt de forma

(1) $AX\{\geqslant, =, \leqslant\}\ b,$

(2) $X \geqslant 0,$

A, b, fiind matrice, respectiv vector cu componente, constante, şi fie funcţia obiectiv

$$f(X) = CX$$

pentru care coeficienţii c_j sînt variabile aleatoare.

Putem aborda problema în mai multe moduri.

Cel mai comod din punct de vedere al tehnicii de calcul, ar consta în considerarea coeficienţilor c_j constanţi, şi anume, egali cu valoarea lor medie (dată de repartiţia lor probabilistică).

Am fi astfel conduşi la o problemă deterministică de programare

Scanned with OKEN Scanner

liniară în care se cere maximizarea sau minimizarea funcției obiectiv

$$f'(X) = M(C)X$$

în condițiile (1), (2); (am notat cu $M(C)$ vectorul $(M(c_1), M(c_2), \ldots, M(c_n))$ ale cărui componente sînt valorile medii ale variabilelor aleatoare $c_1, c_2, \ldots, c_n$).

O asemenea tratare a problemei își are justificarea în interpretarea probabilistică a valorii medii ca indicator al punctului de concentrare al masei probabilistice, pentru valorile unei variabile aleatoare, și poate duce la rezultate corecte (în anumite limite) în situațiile în care modelul concret al problemei indică independența variației coeficienților funcției obiectiv față de variabilele x_j.

În exemplul din paragraful precedent, se explică motivele pentru care o astfel de tratare a problemei, se dovedește, de multe ori, nesatisfăcătoare.

O a doua modalitate de formulare a problemei, mult mai adecvată caracterului real, aleator, al fenomenului modelat constă în optimizarea funcției

$$f''(X) = M(f(X))$$

ceea ce conduce tot la o problemă deterministică, dar, în general, neliniară, așa cum am văzut în exemplul 1 din capitolul III, § 1 (de notat că $f'(X) \neq f''(X)$, egalitatea neavînd loc decît în ipoteza, anterior menționată, a independenței variației vectorului C în raport cu variabilele x_j).

În sfîrșit, o abordare corectă a problemei trebuie să țină seama complet de comportarea aleatoare a variabilelor c_j, ceea ce conferă programului un caracter stochastic.

Este evident, că din natura aleatoare a coeficienților funcției obiectiv, va rezulta și o comportare aleatoare a soluției optime a problemei, cît și a valorii optime a funcției obiectiv.

Pentru un ω dat, problema revine la aflarea maximului (minimului) funcției $C(\omega)X$ în condițiile (1), (2).

Fie $X^*(\omega)$, $C(\omega)X^*(\omega)$ soluția optimă și valoarea optimă a lui f asociată ei, pentru programul liniar de mai sus.

Putem vorbi, atunci, de variabilele aleatoare X^* și CX^* care descriu variația aleatoare a soluției optime, respectiv, a optimului problemei.

Scanned with OKEN Scanner

Ce înţelegem prin rezolvarea programului stochastic?

Determinarea valorii vectorului X^*, pentru care CX^* are cea mai mare valoare posibilă pentru toate realizările ω nu constituie, bineînţeles, un răspuns corect. Aceasta pentru că nu ţine cont de probabilităţile asociate diferitelor valori ale variabilelor aleatoare c_j, şi, implicit, ale lui X^*.

În realitate, fiecare punct extrem al mulţimii convexe a soluţiilor posibile, poate constitui soluţie optimă pentru o mulţime de programe liniare de forma

$$\max(\min)\ C(\omega)X, \tag{3}$$

$$AX\{\geqslant, =. \leqslant\}b, \tag{4}$$

$$X \geqslant 0 \tag{5}$$

cu ω aparţinînd unei anumite mulţimi de evenimente elementare.

Explicaţia acestui fapt constă în aceea că un program liniar de tipul (3)—(5) realizează soluţia optimă într-un punct extrem al mulţimii soluţiilor posibile. Or, pentru toate programele liniare (3)—(5) mulţimea soluţiilor posibile este aceeaşi, deoarece sistemul de restricţii nu are caracter aleator.

Pe de altă parte, fiecare ω determină un alt set de valori pentru vectorul aleator C şi, implicit, o altă formă pentru funcţia obiectiv f, rezultînd de aici posibilitatea plasării soluţiei optime în diferite puncte extreme.

În concluzie, rezolvarea problemei de programare stochastică, cere determinarea repartiţiei probabilistice a vectorului X^*, repartiţie care va fi complet determinată prin indicarea tuturor punctelor extreme ale lui $\mathfrak{X}$ şi prin calcularea probabilităţilor cu care fiecare din acestea devine soluţie optimă a programelor liniare (3)—(5).

Este evident faptul că repartiţia lui X^* este determinată, printre altele, de repartiţiile variabilelor aleatoare c_j.

Pentru a urmări modul în care repartiţia variabilelor c_j induce repartiţia lui X^* să analizăm următoarul exemplu:

Exemplul I. În condiţiile:

$$\begin{aligned} &x_1 + 2x_2 \geqslant 2,\\ &2x_1 - x_2 \geqslant -1,\\ &x_1 + 4x_2 \leqslant 13,\\ &2x_1 - 3x_2 \leqslant 4,\\ &x_1 \geqslant 0,\ i = 1,2, \end{aligned}$$

Scanned with OKEN Scanner

să se maximizeze

$$f(x) = c_1x_1 + c_2x_2,$$

Unde c_1, c_2 sînt două variabile aleatoare independente, urmînd repartiţii uniforme pe intervalele $[-1,4]$, respectiv $[-2,3]$.

În figura 20 patrulaterul $ABCD$, reprezintă mulţimea soluţiilor posibile.

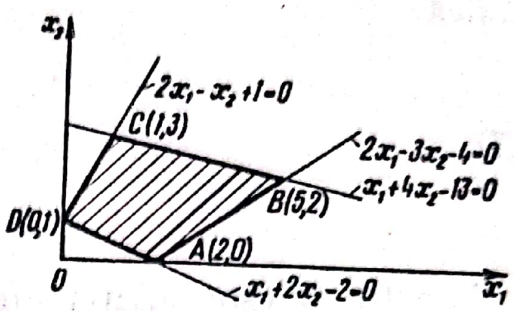


Fig. 20

Valoriile funcţiei obiectiv, în cele patru puncte extreme, sînt

$$f(A) = 2c_1,\ f(B) = 5c_1 + 2c_2,\quad f(C) = c_1 + 2c_2,\ f(D) = c_2$$

şi sînt, bineînţeles, ele însile, variabile aleatoare.

Cum soluţia optimă a programului pentru orice ω, este atinsă, neapărat, într-un asemenea punct extrem, este suficient să punem condiţia ca valoarea lui f într-unul din cele patru puncte să fie superioară valorilor corespunzătoare celorlalte trei.

Astfel, A reprezintă soluţia optimă a tuturor acelor programe, corespunzînd evenimentelor elementare ω pentru care

$$f(A) \geqslant \begin{cases} f(B), \\ f(C), \\ f(D), \end{cases}$$

ceea ce, conduce la sistemul de inecuaţii

$$\begin{aligned} &-3c_1 - 2c_2 \geqslant 0, \\ &c_1 - 3c_2 \geqslant 0, \\ &2c_1 - c_2 \geqslant 0. \end{aligned} \tag{6}$$

În mod asemănător, pentru ca B să fie soluţia optimă este necesar ca

$$f(B) \geqslant \begin{cases} f(A), \\ f(C), \\ f(D) \end{cases}$$

Scanned with OKEN Scanner

adică

$$\begin{aligned} 3c_1 + 2c_2 &\geqslant 0, \\ 4c_1 - c_2 &\geqslant 0, \\ 5c_1 + c_2 &\geqslant 0. \end{aligned} \tag{7}$$

Punctul C este soluția optimă dacă

$$f(C) \geqslant \begin{cases} f(A), \\ f(B), \\ f(D) \end{cases}$$

sau

$$\begin{aligned} -c_1 + 3c_2 &\geqslant 0, \\ -4c_1 + c_2 &\geqslant 0, \\ c_1 + 2c_2 &\geqslant 0. \end{aligned} \tag{8}$$

În sfîrșit, pentru ca punctul D să realizeze maximul lui f, este necesar

$$f(D) \geqslant \begin{cases} f(A), \\ f(B), \\ f(C) \end{cases}$$

sau

$$\begin{aligned} c_2 - 2c_1 &\geqslant 0, \\ -5c_1 - c_2 &\geqslant 0, \\ -c_1 - 2c_2 &\geqslant 0. \end{aligned}$$

Ținînd seama de ipotezele făcute asupra repartițiilor variabilelor aleatoare c_1, c_2, putem da o imagine grafică, sugestivă, a situațiilor discutate aici.

Astfel, fiecare valoare posibilă a vectorului $C = (c_1, c_2)$, poate fi reprezentată printr-un punct în pătratul $[-1,4] \times [-2,3]$, iar probabilitatea fiecărei mulțimi de valori ale vectoru-

Scanned with OKEN Scanner

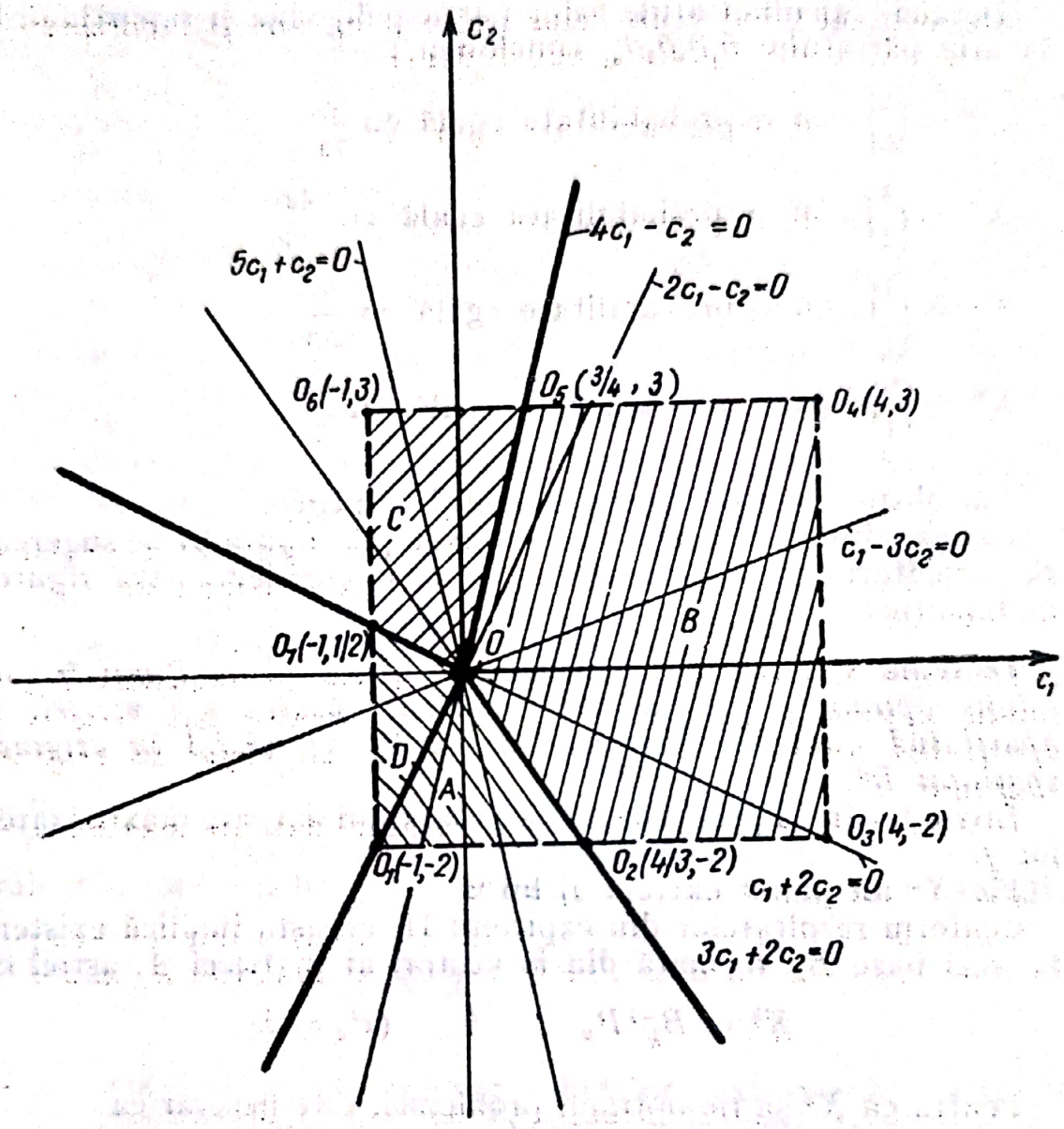


Fig. 21.

lui aleator C, este egală cu raportul dintre aria imaginii acestei mulțimi din pătratul menționat, și aria totală a acestuia.

Pe figura 21, am reprezentat mulțimile de funcții ce reprezintă soluțiile sistemelor (6), (7), (8) și (9).

Pentru ca A, B, C, sau D să fie soluții optime, trebuie ca punctul de coordonate (c_1, c_2) să se găsească în unul din poligoanele OO_1O_2, $OO_2O_3O_4O_5$, $OO_5O_6O_7$, respectiv, OO_7O_1.

Scanned with OKEN Scanner

De aici, calculînd ariile celor patru poligoane și raportîndu-le la aria pătratului $0_1 0_3 0_4 0_6$, concludem:

$X^* = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$, cu o probabilitate egală cu $\frac{7}{75}$,

$X^* = \begin{pmatrix} 5 \\ 2 \end{pmatrix}$, cu o probabilitate egală cu $\frac{421}{600}$,

$X^* = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$, cu o probabilitate egală cu $\frac{31}{200}$,

$X^* = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, cu o probabilitate egală cu $\frac{1}{20}$.

Am obținut astfel repartiția soluției optime.

În baza discuției făcute pe acest exemplu, figura 21 ne sugerează, următorul rezultat general, ce se poate demonstra riguros matematic:

Teorema 1. *Orice vector X, punct extrem al mulțimii $\mathfrak{X}$ este soluție optimă a programului stochastic, pentru toți vectorii C aparținînd unui con poliedric convex Γ_x, cu vîrful în originea spațiului R^n.*

Într-adevăr, să presupunem că programul dat, are maximizarea lui f.

Fie X^k un punct extrem al lui $\mathfrak{X}$.

Conform rezultatelor din capitolul II, aceasta implică existența unei baze B_k, formată din m vectori ai matricei A, astfel ca

$$X^k = B_k^{-1} P_0 \qquad (P_0 = b).$$

Pentru ca X^k să fie optimul problemei, este necesar ca

$$z_j - c_j \geqslant 0, \qquad j = 1,2,\ldots,n$$

unde

$$z_j = C_{B_k} Z_j = C_{B_k} \cdot B_k^{-1} P_j,$$

P_j fiind coloana j a matricei A.

Inegalitățile

$$C_{B_k} \cdot B_k^{-1} P_j - c_j \geqslant 0, \qquad j = 1,2,\ldots,n \tag{10}$$

Scanned with OKEN Scanner

determină însă un con poliedric convex în spaţiul R^n, ceea ce, demonstrează teorema.

În baza acestui rezultat, se poate deduce uşor şi posibilitatea adaptării algoritmului simplex în scopul determinării repartiţiei lui X^*.

Rezultă de asemenea

$$P(X^* = X) = P(C \in \Gamma_x) = \int_{\Gamma_x} dF_C(c_1, c_2, \ldots, c_n) \tag{11}$$

unde $F_C(c_1, c_2, \ldots, c_n)$ reprezintă funcţia de repartiţie a vectorului aleator $C = (c_1, c_2, \ldots, c_n)$, iar Γ_x se determină conform relaţiilor (10).

Egalitatea (11), determină vectorul de probabilitate al variabilei aleatoare X^*, care datorită proprietăţilor mulţimii $\mathfrak{X}$, este de tip discret.

Este mai dificil de exprimat repartiţia valorii optime $f(X^*)$, care, în general, este o variabilă aleatoare de tip continuu.

Putem cita următoarele rezultate:

Teorema 2. *Funcţia* $z^*(C) = \max\{f(X) = CX | AX \leqslant b, X \geqslant 0\}$ *este o funcţie convexă de argument* C.

Funcţia $z^*(C) = \min\{f(X) = CX | AX \geqslant b, X \geqslant 0\}$ *este o funcţie concavă de argument* C.

Cunoscînd repartiţia F_C a vectorului aleator C, putem obţine valoarea medie a lui $f(X^*)$, ţinînd seamă de teorema 1. Rezultă

$$M(CX^*) = \sum_k \int_{\Gamma_{xk}} CX^k \, dF_C(c_1, c_2, \ldots, c_n), \tag{12}$$

suma acţionînd asupra tuturor punctelor extreme X^k ale lui $\mathfrak{X}$.

Se poate demonstra uşor, folosind inegalităţi de tip Jensen, următoarea consecinţă, de interes practic, a teoremei 2:

$$\begin{aligned} M\ (\max\ CX) &\geqslant \max\ (M(C))X, \\ M\ (\min\ CX) &\leqslant \min\ (M(C))X. \end{aligned} \tag{13}$$

Inegalităţile (13) arată că rezolvarea problemei stochastice duce, în medie, la rezultate mai bune decît rezolvarea programului deterministic obţinut prin înlocuirea vectorului aleator C cu valoarea sa medie.

Scanned with OKEN Scanner

Metoda de rezolvare a programului sugerată de teorema 1, şi folosită în forma grafică în exemplul studiat, se bazează pe algoritmul simplex şi constă în descrierea completă a repartiţiei lui X^*. Dat fiind, însă, numărul mare al punctelor extreme ce apar în problemele practice, metoda devine laborioasă.

Trebuie menţionat de asemenea, că din punct de vedere practic, pentru adoptarea unei decizii trebuie să se ţină seama de o serie de alte criterii. Într-adevăr, adoptarea unei decizii implică alegerea unui singur vector X drept politică în fenomenul modelat.

Care va fi sensul noţiunii de optim pe care putem să-l atribuim acestei politici? Cunoaşterea repartiţiei probabilistice a lui X^*, nu indică decît posibilele soluţii optime, supuse legilor hazardului. Pe de altă parte, chiar pentru fiecare poziţie posibilă a punctului X^*, trebuie să se ţină seamă de multitudinea de valori posibile ale lui f.

Un răspuns adecvat întrebării puse aici nu se poate da, decît, îmbunătăţind rezultatele generale, expuse anterior, cu observaţii legate de natura concretă a problemei, elaborîndu-se astfel o „strategie" corespunzătoare.

§ 3. PROGRAME STOCHASTICE CU VECTORUL b ALEATOR

Fără a restrînge generalitatea, să fixăm sensul inegalităţilor în restricţii precum şi natura cerinţei de optim şi să considerăm programul adus la forma:

În condiţiile

$$AX \geqslant b, \tag{1}$$

$$X \geqslant 0, \tag{2}$$

să se determine

$$\text{minim } \{f(X) = CX\} \tag{3}$$

unde A, C, sînt matrice, respectiv vector linie cu elemente constante, iar b este un vector m — dimensional aleator.

Ca şi în cazul precedent, problema poate fi abordată într-o manieră deterministică, înlocuindu-l pe b cu vectorul $M(b)$,

Scanned with OKEN Scanner

ale cărui componente sînt valorile medii $M(b_i)$, $(i = 1,2,\ldots,m)$ ale variabilelor aleatoare b_i.

Mai mult, în acest caz vom arăta că asemenea tratare a problemei se dovedeşte, într-un anume sens, mai avantajoasă.

Dacă ţinem seama însă de variaţia aleatoare a vectorului b problema dată apare ca un program stochastic. Evident, caracterul aleator pe care îl prezintă o parte a parametrilor, va imprima, şi de această dată, o comportare aleatoare soluţiei problemei.

Ne propunem să deducem repartiţia de problabilitate a optimului, folosind, ca şi în paragraful precedent, metoda descrierii complete.

Spre deosebire de cazul precedent cînd elementul aleator apărea doar în expresia funcţiei obiectiv, de data aceasta caracterul aleator al vectorului b implică variabilitatea mulţimii soluţiilor posibile şi deci şi a punctelor extreme ale acesteia.

Putem observa, totuşi, că deoarece matricea A rămîne constantă, în efect, posibilele baze optime vor rămîne şi ele invariabile.

Ca atare, nu rămîne decît să stabilim care sînt aceste baze optimale, şi ţinînd seama de repartiţia vectorului aleator b să determinăm probabilităţile cu care fiecare din bazele astfel selecţionate devine optimă.

De remarcat, că pentru o asemenea bază optimă, soluţia ataşată ei nu este unică, ea variind odată cu b.

Într-adevăr, dacă B este o bază a programului care pentru o anumită mulţime de evenimente ω devine optimă, soluţia optimă a problemei se obţine rezolvînd sistemul liniar

$$BX = b(\omega)$$

pentru fiecare ω în mulţimea considerată. Or, cum valorile lui b diferă de la un ω la altul, este normal ca şi cu X^* să se întîmple acelaşi lucru. De aici concluzia că X^* va fi o variabilă aleatoare de tip continuu.

Acest fapt face, uneori, imposibilă specificarea completă a repartiţiei lui X^*. Putem însă pune problema determinării repartiţiei bazei optime B^*, plecînd de la legea probabilitistică a vectorului aleator b.

Fie atunci programul dual corespunzător problemei date:

(4) Să se maximizeze

$$g(W) = Wb,$$

Scanned with OKEN Scanner

cu restricţiile :

$$WA \leqslant C', \tag{5}$$

$$W \geqslant 0. \tag{6}$$

Evident programul dual mută parametrii aleatori din sistemul de restricţii, în expresia funcţiei de eficienţă, obţinînd astfel o problemă de tipul celor studiate în § 2.

Pentru a vedea modul în care putem utiliza rezultatele obţinute pentru acest tip de probleme în cazul de faţă, să analizăm un exemplu.

Exemplul I.

$$-x_1 - 2x_2 + x_3 + 2x_4 \geqslant b_1,$$

$$-2x_1 + x_2 + 4x_3 - 3x_4 \geqslant b_2,$$

$$x_i \geqslant 0,\ i = 1, \ldots,4;$$

$$\min (2x_1 + x_2 + 13x_3 + 4x_4)$$

b_1 şi b_2 fiind variabile aleatoare independente urmînd repartiţile uniforme pe intervalul $[-1,4]$, respectiv $[-2,3]$.

Pentru aplicarea metodei simplex punem sistemul de restricţii sub forma

$$-x_1 - 2x_2 - x_3 + 2x_4 - y_1 + x_5^a = b_1,$$

$$-2x_1 + x_2 + 4x_3 - 3x_4 - y_2 + x_6^a = b_2$$

În tabelul 47 am căutat să punem în evidenţă condiţiile în care baza formată din vectorii P_4 şi P_3 poate fi optimă.

Deoarece toţi $z_j - c_j \leqslant 0$, optimul a fost atins, cu condiţia ca să fi fost respectate toate prescripţiile metodei.

În prima parte a tabelului 47 pentru ca P_2 să intre în bază rezultă, din regula de alegere a pivotului, necesitatea condiţiei

$$b_2 \geqslant 0. \tag{I}$$

În partea a doua a tabelului 47, din motive identice, P_4 intră în bază dacă

$$4b_1 - b_2 \geqslant 0. \tag{II}$$

Scanned with OKEN Scanner

Tabelul 47

C_B	c_j	$\rightarrow$	-2	1	13	4	0	0	α	α
$\downarrow$	variabile bazice	P_0	P_1	P_2	P_3	P_4	Q_1	Q_2	η_1	η_2
α	x_5^a	b_1	-1	-2	1	2	-1	0	1	0
α	x_6^a	b_2	-2	1	$\boxed{4}$	-3	0	-1	0	1
	$z_j - c_j$									
	$z_j - c_j'$		3α	$-\alpha$	$+5\alpha$	$-\alpha$	$-\alpha$	$-\alpha$	0	0
α	x_5^a	$\frac{4b_1 - b_2}{4}$	$\frac{1}{2}$	$-5/4$	0	$\boxed{11/4}$	-1	$1/4$	1	1
13	x_3^a	$\frac{b_2}{4}$	$\frac{-1}{2}$	$1/4$	1	$-3/4$	0	$-1/4$	0	
	$z_j - c_j$									
	$z_j - c_j'$		$\alpha/2$	$-5\alpha/4$	0	$11\alpha/4$	$-\alpha$	$\alpha/4$	0	
4	x_4	$\frac{4b_1 - b_2}{11}$	$-2/11$	$-5/11$	0	1	$-4/11$	$1/11$		
13	x_3	$\frac{3b_1+2b_2}{11}$	$-7/11$	$-1/11$	1	0	$-3/11$	$-2/11$		
	$z_j - c_j$		-7	-4	0	0	-5	-2		

Scanned with OKEN Scanner

În sfîrşit pentru ca ultima parte a tabelului să fie o soluţie optimă (deci posibilă) este necesar ca toate componentele acestuia să fie nenegative. Aceasta, ţinînd seama de (II) dă condiţia

$$3b_1 + 2b_2 \geqslant 0. \tag{III}$$

În concluzie, baza formată de P_4 şi P_3 este optimă pentru acei ω, pentru care (I), (II), (III) au loc.

Să asociem problemei date duala:

$$\max (b_1 w_1 + b_2 w_2)$$

în condiţiile

$$\begin{aligned} -w_1 - 2w_2 &\leqslant -2, \\ -2w_1 + w_2 &\leqslant 1, \\ w_1 + 4w_2 &\leqslant 13, \\ 2w_1 - 3w_2 &\leqslant 4, \\ w_i &\geqslant 0, \qquad i = 1,2 \end{aligned}$$

Făcînd abstracţie de deosebirile de notaţie, această problemă coincide cu cea din exemplul 1 din § 2.

Din rezolvarea acestui exemplu rezultă că $W^* = (2, 0)$, dacă vectorul b aparţine domeniului $00_2 0_3 0_4 0_5$ (fig. 21).

Or, punctului (2,0) îi corespund în problema duală liniile a treia şi a patra, şi deci, în problema iniţială baza formată din vectorii P_4 şi P_3.

Pe de altă parte, domeniul $00_2 0_3 0_4 0_5$ coincide cu domeniul generat de inegalităţile (I) — (III), dacă ţinem seama de repartiţiile variabilelor aleatoare b_1, b_4.

Observaţii asemănătoare se pot face şi în legătură cu celelalte baze optime ale programului considerat.

Folosind rezultatele exemplului 1 din § 2, obţinem repartiţia bazei optime B, anume:

$$\begin{aligned} B^* &= (P_1, P_4), \text{ dacă } b = (b_1, b_2) \in 00_1 0_2, \\ B^* &= (P_4, P_3), \text{ dacă } b \in 00_2 0_3 0_4 0_5, \\ B^* &= (P_3, P_2), \text{ dacă } b \in 00_5 0_6 0_7, \\ B^* &= (P_1, P_2) \text{ dacă } b \in 00_7 0_1 \end{aligned}$$

și deci, probabilitățile celor patru valori posibile ale bazei optime vor fi $\frac{7}{75}$, $\frac{421}{600}$, $\frac{31}{200}$ și respectiv $\frac{1}{20}$.

Plecînd de la observațiile făcute pe acest exemplu particular, putem formula următorul rezultat general:

Teorema 1. *Un sistem de m vectori coloană ai matricei A, P_{i_1}, $P_{i_2}, \ldots, P_{i_m}$ reprezintă o bază optimă a programului stochastic* (1)—(3), *pentru toți vectorii b aparținînd unui con convex cu vîrful în originea spațiului R^m.*

Acest con se poate obține folosind teorema 1 (§ 2), în problema dual㠠(4)—(6). Dăm, de asemenea fără demonstrație și următoarea teoremă:

Teorema 2. *Funcția* $z^*(b) = \min \{CX|AX \geqslant b, X \geqslant 0\}$ *este o funcție convexă de argument b.*

Funcția $z^*(b) = \max \{CX|AX \leqslant b, X \geqslant 0\}$ *este o funcție concavă de b.*

Ca o consecință a ultimei teoreme, se pot demonstra ușor inegalitățile:

$$(7) \qquad M(\max\{CX|AX \leqslant b, X \geqslant 0\}) \leqslant \max\{CX|AX \leqslant M(b), X \geqslant 0\},$$

$$M(\min\{CX|AX \geqslant b, X \geqslant 0\}) \geqslant \min\{CX|AX \geqslant M(b), X \geqslant 0\}.$$

Aici, valoarea medie a optimului funcției obiectiv se scrie, ținînd seama de teorema 1 și de rezultatul analog obținut în § 2,

$$M(CX^*) = M(W^*b) = \sum_k \int_{\Gamma_{W^k}} W^k b \, dF_b(b_1, b_2, \ldots, b_m)$$

suma extinzîndu-se asupra tuturor punctelor extreme ale mulțimilor soluțiilor posibile ale dualei (F_b este funcția de repartiție a vectorului aleator b).

Din relațiile (7) se desprinde ideia, că spre deosebire de cazul programelor stochastice cu element aleator C, în problemele ce presupun variabilitatea lui b, este avantajoasă înlocuirea

variabilelor aleatoare b_i $(i = 1,2,\ldots,m)$ cu valorile lor medii $M(b_i)$.

În felul acesta valoarea optimă a funcţiei obiectiv este mai bună decît media valorilor optime corespunzătoare programului stochastic.

În încheiere, să subliniem faptul că metoda descrierii complete aplicată programului (1)—(3), prezintă, pe lîngă dezavantajul volumului mai mare de calcul şi pe cel ce decurge din imposibilitatea explicitării soluţiei optime.

Bineînţeles, rezolvarea programului stochastic trebuie să se facă într-un cadru mult mai complex, în conexiune cu o serie de alte considerente, ţinînd de natura aleatoare concretă a fenomenului conferindu-i astfel caracterul unui proces de decizie.

Scanned with OKEN Scanner

Scanned with OKEN Scanner

Capitolul VII

PROGRAMAREA DINAMICĂ

Termenul de „programare dinamică" defineşte o tehnică de calcul ce poate fi folosită în rezolvarea unor anumite tipuri de probleme de optim, în particular, şi a unor probleme de programare matematică.

Ideile ce au fundamentat procedeele programării dinamice se datoresc lui R. Bellman.

Vom încerca, în cele ce urmează, să expunem, pe scurt, aceste idei, pe unele cazuri particulare simplificatoare.

§ 1. PRINCIPIUL OPTIMALITĂŢII

Presupunem dată o problemă de decizie care cere optimizarea unei anumite funcţii prin alegerea convenabilă a valorilor variabilele $x_1, x_2, \ldots, x_N$ supuse unui sistem de restricţii.

Valorile optime ale variabilelor *de control* $x_1, x_2 \ldots, x_N$ precum şi valoarea optimă a funcţiei date, vor depinde de un sistem de parametri, numiţi *parametrii de stare*, care descriu starea sistemului, adică determină sistemul de restricţii la care sînt supuse variabilele.

Procedeele programării dinamice se aplică acelor probleme ce pot fi formulate ca procese de decizie cu mai multe stadii, fiecăruia dintre acestea fiindu-i proprii una sau mai multe variabile de control.

Totodată este necesar ca problemele de decizie corespunzătoare diferitelor stadii să aibă o structură asemănătoare.

Dacă aceste condiţii sînt îndeplinite atunci problema dată poate fi formulată ca o problemă de programare dinamică, care presupune rezolvarea succesivă a unui număr, de n subprobleme, începînd cu una ce corespunde primului stadiu al procesului, apoi cu una care să corespundă primelor două stadii, culminînd cu ultima care va corespunde tuturor stadiilor şi care va da astfel soluţia problemei originare.

Rezolvarea problemei puse în etapa k ($k \leqslant n$), se face adăugînd stadiul k problemei cu $k-1$ stadii rezolvată la etapa precedentă şi făcînd uz de soluţia obţinută pentru aceasta.

Valorile date variabilelor de control proprii stadiului k, în problema de decizie ce ia în considerare primele k stadii, nu trebuie să aibă alt efect asupra primelor $k-1$ stadii, decît schimbarea parametrilor de stare corespunzători acestora.

Să notăm cu X_k vectorul format cu variabilele de control corespunzătoare stadiului k şi cu ξ_k vectorul parametrilor de stare corespunzători problemei de decizie care se referă la primele k stadii. Rezolvînd problema pusă la această etapă, fie X_k^* valoarea optimă a lui X_k şi $f_k(X_k^*, \xi_k)$ valoarea optimă a funcţiei obiectiv corespunzătoare primelor k etape.

Una din ideile de bază ale programării dinamice constă în a determina o relaţie de recurenţă simplă din care să poată rezulta $f_k(X_k^*, \xi_k)$, cunoscînd decizia optimă luată în problema cu $k-1$ stadii rezolvată în etapa precedentă.

O asemenea relaţie tipică proceselor de programare dinamică este

$$(1)\quad f_k(X_k^*, \xi_k) = \underset{X_k}{\text{optim}} \{g_k(X_k, \xi_k) + f_{k-1}[X_{k-1}^*, T(\xi_k, X_k)]\}$$

unde $T(\xi_k, X_k)$ reprezintă vectorul de stare corespunzător etapei $k-1$, rezultat în urma adoptării deciziei X_k.

În ultima etapă (a n-a) se obţin X_n^* şi $f_n(X_n^*, \xi)$, ξ reprezentînd vectorul parametrilor de stare corespunzător problemei originare.

Vectorul $(X_1^*, X_2^*, \ldots, X_n^*)$, reprezentînd deciziile optime în cele n etape ale procesului, îl vom numi *politică optimă* sau *strategie optimă* pentru cele n etape.

Procedeele programării dinamice sînt dominate de *principiul optimalităţii* potrivit căruia:

Scanned with OKEN Scanner

Oricare ar fi decizia X_{k-1} luată în etapa $k-1$, corespunzător stării ξ_k rezultată din aceasta, politica $(X_k^*, X_{k+1}^* \ldots, X_n^*)$ trebuie să fie optimă pentru etapele rămase.

În general, în problemele ce se pretează a fi rezolvate cu procedeele programării dinamice, divizarea în etape nu decurge dintr-un imperativ obiectiv, ci este făcută de analist, în scopul aplicării procedeelor corespunzătoare.

Sînt însă unele procese, pe care le vom numi *secvențiale*, în care există o ordine temporală ce se impune deciziilor parțiale corespunzătoare etapelor procesului, ordine care va trebui să se resfrîngă și asupra etapelor ce apar în rezolvarea dinamică a problemei.

Pentru procesele de decizie secvențiale, în programarea dinamică se pot folosi două tipuri de soluții : *progresive** și *regresive***.

O soluție progresivă respectă ordinea temporală a procesului, prima etapă a programului dinamic dînd prima decizie ce trebuie să fie luată. În cadrul unei soluții regresive, prima etapă a programului dinamic dă ultima decizie ce trebuie luată, în timp.

În exemplul din § 3, vom da cîte o soluție din fiecare tip.

§ 2. TEHNICA DE CALCUL A PROGRAMĂRII DINAMICE

Să considerăm problema de programare matematică :

$$(1)\quad \sum_{j=1}^{n} a_{ij}x_j \leqslant b_i, \quad a_{ij} > 0, \quad j = 1, 2, \ldots, n, \; i = 1, 2, \ldots, m;$$

$$(2)\quad x_j \geqslant 0, \qquad j = 1, 2, \ldots, n;$$

$$(3)\quad \max \sum_{j=1}^{n} g_j(x_j).$$

Fie $z^* = \max \sum_{j=1}^{n} g_j(x_j)$, în condițiile (1) și (2).

---

* forward.
** backward.

Scanned with OKEN Scanner

Să presupunem că am fixat o valoare $\bar{x}_n$ pentru x_n.

În acest caz valorile variabilelor $x_1, x_2, \ldots, x_{n-1}$ care maximizează funcţia obiectiv $\sum_{j=1}^{n} g_j(x_j)$ vor depinde evident de $\bar{x}_n$.

Cu $x_n = \bar{x}_n$, putem lua în considerare programul care cere:

$$(3') \qquad \max_{x_1, x_2, \ldots, x_{n-1}} \sum_{j=1}^{n} g_j(x_j) = g_n(\bar{x}_n) + \max_{x_1, x_2, \ldots, x_{n-1}} \sum_{j=1}^{n-1} g_j(x_j).$$

Este clar, că variabilele $x_1, x_2, \ldots, x_{n-1}$, vor trebui să satisfacă sistemul de restricţii (2), (4) unde (4) este

$$(4) \qquad \sum_{j=1}^{n-1} a_{ij} x_j \leqslant b_i - a_{in}\bar{x}_n, \qquad i = 1, 2, \ldots, m.$$

Notînd cu a_n vectorul coloană de componente $a_{in} (i = 1, 2, \ldots, \ldots, m)$, şi cu b vectorul de componente b_i, să punem

$$f_{n-1}(b - \bar{x}_n a_n) = \max_{x_1, x_2, \ldots, x_{n-1}} \sum_{j=1}^{n-1} g_j(x_j)$$

maximul fiind calculat în condiţiile (2), (4).

Totodată, din (1) şi (2), rezultă că $\bar{x}_n$ trebuie ales din mulţimea

$$(5) \qquad 0 \leqslant x_n \leqslant \min_{1 \leqslant i \leqslant m} \frac{b_i}{a_{in}}.$$

Este clar atunci că

$$(6) \qquad z^* = \max_{x_n} [g_n(x_n) + f_{n-1}(b - x_n a_n)]$$

maximul fiind calculat în condiţiile (5).

Observăm deci, că dacă am putea obţine pe $f_{n-1}(b - x_n a_n)$, atunci problema originară s-ar reduce la rezolvarea programului (5), (6), care conţine doar o singură variabilă.

Scanned with OKEN Scanner

Pentru a vedea care este modalitatea de a obține pe $f_{n-1}(b - x_n a_n)$, să observăm că aceasta reprezintă valoarea maximă a funcției $\sum_{j=1}^{n-1} g_j(x_j)$ în condițiile

$$(4') \qquad \sum_{j=1}^{n-1} a_{ij} x_j \leqslant b_j - a_{in} x_n, \qquad i = 1, 2, \ldots, m,$$

$$(1') \qquad x_j \geqslant 0, \qquad j = 1, 2, \ldots, n-1.$$

Procedînd în același mod ca în etapa precedentă, va rezulta

$$(7) \quad f_{n-1}(b - x_n a_n) = \max_{x_{n-1}} [g_{n-1}(x_{n-1}) + f_{n-2}(b - x_n a_n - x_{n-1} a_{n-1})]$$

în condițiile

$$(8) \qquad 0 \leqslant x_{n-1} \leqslant \min_{1 \leqslant i \leqslant m} \frac{b_i - a_{in} x_n}{a_{in-1}}$$

și unde

$$f_{n-2}(b - x_n a_n - x_{n-1} a_{n-1}) = \max_{x_1, \ldots, x_{n-2}} \sum_{j=1}^{n-2} g_j(x_j)$$

variabilele $x_1, \ldots, x_{n-2}$ satisfăcînd sistemul de restricții (1),(9),

$$(9) \quad \sum_{j=1}^{n-2} a_{ij} x_j \leqslant b_i - a_{in} x_n - a_{in-1} x_{n-1}, \qquad i = 1, 2, \ldots, m.$$

Obținem astfel o descompunere a problemei date, într-o succesiune de subprograme.

În general, la etapa k, vom avea de rezolvat programul

$$\max_{x_1, \ldots, x_k} \sum_{j=1}^{k} a_{ij} x_j$$

$$\sum_{j=1}^{k} a_{ij} x_j \leqslant b_i - a_{in} x_n - \ldots - a_{ik+1} x_{k+1},$$

$$x_j \geqslant 0, \qquad j = 1, 2, \ldots, k$$

valoarea optimă a obiectivului fiind $f_k(b_i - \xi_k)$ (am notat $\xi_k = x_n a_n + \ldots + x_{k+1} a_{k+1}$).

Scanned with OKEN Scanner

Dacă în etapa n putem obține valoarea optimă $f_1(b - \xi_1)$, atunci, $f_2(b - \xi_2), \ldots, f_{n-1}(b - \xi_{n-1})$, se pot calcula recursiv, din relația

$$(10) \qquad f_k(b - \xi_k) = \max_{x_k} [g_k(x_k) + f_{k-1}(b - \xi_{k-1})] =$$

$$= \max_{x_k} [g_k(x_k) + f_{k-1}(\xi_k - a_k x_k)]$$

ajungînd în final la $z^* = f_n(b)$, adică la optimul căutat. Totodată (10) ne dă și decizia optimă x_k^*, corespunzătoare fiecărei etape, adică componenta k a politicii optime $(x_1^*, x_2^*, \ldots, x_n^*)$, care nu va fi altceva decît soluția optimă a problemei inițiale.

În cazul cînd

$$g_j(x_j) = c_j x_j, \qquad (c_j \in R),$$

adică cînd programul inițial este liniar, procesul dinamic descris aici se reduce la rezolvarea unei succesiuni de programe liniare.

§ 3. REZOLVAREA UNUI EXEMPLU NUMERIC

Să presupunem că o unitate comercială trebuie să fie în măsură să răspundă unei cereri cunoscute, dar variabile, pe o perioadă de 6 luni.

La începutul fiecărei luni, unitatea poate cumpăra orice cantitate din produsul solicitat la un preț ce diferă de la lună la lună, conform datelor de mai jos:

luna i	1	2	3	4	5	6
cererea c_i	8	5	3	2	7	4
prețul unitar p_i	11	18	13	17	20	10

Scanned with OKEN Scanner

Ţinînd cont de variaţiile preţului de cumpărare, unitatea îşi propune să-şi planifice o politică de cumpărare, dat fiind că poate să deţină în stoc o cantitate de produs care să nu depăşească 9 unităţi. Stocul iniţial este de două unităţi, iar la sfîrşitul perioadei trebuie să fie zero.

Prima rezolvare. Soluţie progresivă

Evident, problema stabilirii politicii optime, pentru aprovizionarea cu produs, în cazul descris, are un caracter temporal.

Să notăm cu x_i, numărul de unităţi de produs ce trebuie cumpărate la începutul lunii $i (i = 1, 2, 3, 4, 5, 6)$.

Întrucît se stipulează necesitatea satisfacerii integrale a cererii, rezultă că unitatea trebuie să deţină la începutul fiecărei luni o cantitate de produs cel puţin egală cu cererea din luna care începe.

Or, cantitatea de produs deţinută de unitate la începutul lunii i, rezultă din diferenţa dintre toate cantităţile de produs cumpărate pînă în acest moment şi cantităţile de produs cerute în precedentele $i - 1$ luni.

Cu alte cuvinte

$$(1) \qquad s_0 + \sum_{j=1}^{i} s_j - \sum_{j=1}^{i-1} c_j \geqslant c_i, \qquad i = 1, 2, \ldots, 5,$$

$$s_0 + \sum_{j=1}^{6} x_j - \sum_{j=1}^{5} c_j = c_6.$$

Am notat în (1), stocul iniţial cu s_0.

Pe de altă parte, cantitatea de produs aflată în unitate nu trebuie să depăşească, în nici un moment, 9 unităţi.

Cum cantitatea maximă de produs, pentru fiecare lună, este aceea de la începutul ei, rezultă

$$(2) \qquad s_0 + \sum_{j=1}^{i} x_j - \sum_{j=1}^{i-1} c_j \leqslant 9, \qquad i = 1, 2, \ldots, 6.$$

Evident, cheltuielile totale de cumpărare pentru politica $(x_1, x_2, \ldots, x)$ sînt date de expresia

$$\sum_{i=1}^{6} p_i x_i.$$

Scanned with OKEN Scanner

Ţinînd seama şi de condiţia naturală $x \geqslant 0$, sîntem conduşi la următorul program liniar, rezultat din (1) şi (2):

$$\min (11x_1 + 18x_2 + 13x_3 + 17x_4 + 20x_5 + 10x_6), \tag{3}$$

$$\begin{aligned}
&6 \leqslant x_1 \leqslant 7,\\
&11 \leqslant x_1 + x_2 \leqslant 15,\\
&14 \leqslant x_1 + x_2 + x_3 \leqslant 20,\\
&16 \leqslant x_1 + x_2 + x_3 + x_4 \leqslant 23,\\
&23 \leqslant x_1 + x_2 + x_3 + x_4 + x_5 \leqslant 25,\\
&x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 27,
\end{aligned} \tag{4}$$

$$x_i \geqslant 0, \qquad i = 1, 2, \ldots, 6. \tag{5}$$

Vom trata această problemă folosind principiile programării dinamice. Descompunerea pe etape o vom face respectînd ordinea naturală, în timp, a procesului, în scopul de a obţine o soluţie progresivă.

Etapa I. Enunţăm un program pe baza primului stadiu al procesului, vizînd variabila de control x_1,

$$(\mathrm{I}_a) \quad \min_{6 \leqslant x_1 \leqslant 7} [11x_1 + \min_{x_2, \ldots, x_6} (18x_2 + 13x_3 + 17x_4 + 20x_5 + 10x_6)].$$

Ca efect, celelalte variabile, vor optimiza programul

$$\begin{aligned}
(\mathrm{I}_b) \quad &\min (18x_2 + 13x_3 + 17x_4 + 20x_5 + 10x_6) = f_1,\\
&11 - x_1 \leqslant x_2 \leqslant 15 - x_1,\\
&14 - x_1 \leqslant x_2 + x_3 \leqslant 20 - x_1,\\
&16 - x_1 \leqslant x_2 + x_3 + x_4 \leqslant 23 - x_1,\\
&23 - x_1 \leqslant x_2 + x_3 + x_4 + x_5 \leqslant 25 - x_1,\\
&x_2 + x_3 + x_4 + x_5 + x_6 = 27 - x_1,\\
&x_i \geqslant 0, \qquad i = 2, \ldots, 6.
\end{aligned}$$

Scanned with OKEN Scanner

Etapa a II-a. Luînd în considerare primele două stadii ale procesului, variabila de control x_2, corespunzătoare stadiului al doilea, va optimiza programul dedus din (I_b):

$$(II_a) \quad \min_{11-x_1 \leqslant x_2 \leqslant 15-x_1} [18x_2 + \min_{x_3,\ldots,x_6} (13x_3 + 17x_4 + 20x_5 + 10x_6)].$$

Pentru variabilele rămase, formulăm problema:

$$(II_b) \qquad \min(13x_4 + 17x_4 + 20x_5 + 10x_6) = f_2,$$

$$14 - x_1 - x_2 \leqslant x_3 \leqslant 20 - x_1 - x_2,$$

$$16 - x_1 - x_2 \leqslant x_3 + x_4 \leqslant 23 - x_1 - x_2,$$

$$23 - x_1 - x_2 \leqslant x_3 + x_4 + x_5 \leqslant 25 - x_1 - x_2,$$

$$x_3 + x_4 + x_5 + x_6 = 27 - x_1 - x_2$$

$$x_i \geqslant 0, \qquad i = 3,\ldots,6.$$

Etapa a III-a. Adăugînd şi al treilea stadiu, primelor două, variabila de control x_3, va trebui să satisfacă împreună cu soluţia optimă a etapei precedente

$$(III_a) \quad \min_{\substack{14-x_1-x_2 \leqslant x_3 \leqslant 20-x_1-x_2 \\ x_3 \geqslant 0}} [13x_3 + \min_{x_4, x_5, x_6} (17x_4 + 20x_5 + 10x_6)]$$

Variabilele rămase trebuie să optimizeze programul:

$$(III_b) \qquad \min\ (17x_4 + 20x_5 + 10x_6) = f_3,$$

$$16 - \sum_{j=1}^{3} x_j \leqslant x_4 \leqslant 23 - \sum_{j=1}^{3} x_j,$$

$$23 - \sum_{j=1}^{3} x_j \leqslant x_4 + x_5 \leqslant 25 - \sum_{j=1}^{3} x_j,$$

$$x_4 + x_5 + x_6 = 27 - \sum_{j=1}^{3} x_j,$$

$$x_i \geqslant 0, \qquad i = 1,2,3.$$

Scanned with OKEN Scanner

Etapa a IV-a. În cadrul problemei corespunzătoare primelor patru stadii, variabila x_4 împreună cu soluția optimă a problemei precedente, satisface condiția

$$(IV_a) \qquad \min_{16-\sum_{j=1}^{3} x_j \leqslant x_4 \leqslant 23-\sum_{j=1}^{3} x_j} [17x_4 + \min_{x_5,\ x_6} (20x_5 + 10x_6)].$$

Ultimele două variabile, vor optimiza programul

$$(IV_b) \qquad \min(20x_5 + 10x_6) = f_4,$$

$$23 - \sum_{j=1}^{4} x_j \leqslant x_5 \leqslant 25 - \sum_{j=1}^{4} x_j,$$

$$x_5 + x_6 = 27 - \sum_{j=1}^{4} x_j,$$

$$x_5 \geqslant 0, \qquad x_6 \geqslant 0.$$

Etapa a V-a. Variabila de control x_5, corespunzătoare stadiului al cincilea, se determină din

$$(V_a) \qquad \min_{23-\sum_{j=1}^{4} x_j \leqslant x_5 \leqslant 25-\sum_{j=1}^{4} x_j} (20x_5 + \min_{x_6} 10x_6).$$

Pentru x_6 rezultă

$$(V_b) \qquad \min 10x_6 = f_5,$$

$$x_6 = 27 - \sum_{j=1}^{5} x_j,$$

$$x_6 \geqslant 0.$$

Etapa a VI-a. Ultima etapă constă în rezolvarea problemei (V_b). Deoarece din (4) avem.

$$23 \leqslant \sum_{j=1}^{5} x_j \leqslant 25$$

rezultă posibilitatea rezolvării problemei (V_b).

Scanned with OKEN Scanner

Evident, optimul este atins pentru

$$x_6 = 27 - \sum_{j=1}^{5} x_j. \tag{6}$$

De aici

$$f_5 = 10 \sum_{j=1}^{5} x_j + 270. \tag{7}$$

Folosind (7) în (V_a) rezultă

$$\min_{23 - \sum_{j=1}^{4} x_j \leqslant x_5 \leqslant 25 - \sum_{j=1}^{4} x_j \quad x_5 \geqslant 0} [10x_5 - 10x_4 - 10x_3 - 10x_2 - 10x_1 + 270]. \tag{8}$$

Expresia din paranteză fiind crescătoare în raport cu variabila x_5, minimul va fi atins pentru

$$x_5 = 23 - \sum_{j=1}^{4} x_j. \tag{9}$$

Atunci, valoarea optimă în (IV_b), rezultă din (8) și (9):

$$f_4 = [-20x_4 - 20x_3 - 20x_2 - 20x_1 + 500]. \tag{10}$$

În consecință (IV_a) devine.

$$\min_{16 - \sum_{j=1}^{3} x_j \leqslant x_4 \leqslant 23 - \sum_{j=1}^{3} x_j} [-3x_4 - 20x_3 - 20x_2 - 20x_1 + 500] \tag{11}$$

Cum expresia din paranteză este descrescătoare în raport cu variabila x_4, soluția optimă este

$$x_4 = 23 - \sum_{j=1}^{3} x_j, \tag{12}$$

Scanned with OKEN Scanner

Din (12), (11) și (III_b) rezultă

$$f_3 = [-17x_3 - 17x_2 - 17x_1 + 431]. \tag{13}$$

Atunci (III_a) devine

$$\min_{\substack{14-x_1-x_2 \leqslant x_3 \leqslant 20-x_1-x_2 \\ x_3 \geqslant 0}} [-4x_3 - 17x_2 - 17x_1 + 431]. \tag{14}$$

Minimul este atins pentru

$$x_3 = 20 - x_1 - x_2. \tag{15}$$

Corespunzător acestei valori, din (II_b) și (14) avem

$$f_2 = [-13x_2 - 13x_1 + 351]. \tag{16}$$

Și problema (II_a) devine

$$\min_{11-x_1 \leqslant x_2 \leqslant 15-x_1} [5x_2 - 13x_1 + 351]. \tag{17}$$

Funcția obiectiv fiind crescătoare în raport cu x_2, minimul este luat pentru

$$x_2 = 1 - x_1. \tag{18}$$

În sfîrșit, (18), (17) și (I^b) dau

$$f_1 = [-18x_1 + 406]. \tag{19}$$

De aici, (I_a) devine

$$\min_{6 \leqslant x_1 \leqslant 7} [-7x_1 + 406]. \tag{20}$$

Soluția evidentă este

$$x_1^* = 7.$$

De aici, înlocuind succesiv în (18, (15), (12), (9) și (6) se obține

$$x_2^* = 4, \quad x_3^* = 9, \quad x_4^* = 3, \quad x_5^* = 0, \quad x_6^* = 4.$$

În concluzie, politica optimă este

$$(7,4,9,3,0,4).$$

Aceasta corespunde soluției optime a problemei puse inițial. Minimul cheltuielilor totale este dat de (20)

$$z^* = 357.$$

A doua rezolvare. Soluție regresivă

Putem da o altă formulare problemei puse, alegînd alte variabile de control.

Să notăm astfel cu $y_i, (i = 1,2,\ldots,6)$, stocul de produs ce se găsește în unitate la începutul lunii i, după aprovizionarea cu produs.

Legătura dintre aceste noi variabile și variabilele x_i, folosite anterior, rezultă imediat

$$y_1 = x_1 + s_0,$$

(21)

$$y_i = x_i + y_{i-1} - c_{i-1}, \qquad i = 2,\ldots,6.$$

Într-adevăr stocul la începutul lunii i rezultă din cantitatea de produs x_i cumpărată la începutul acestei luni, și stocul rămas la sfîrșitul lunii precedente, adică tocmai $y_{i-1} - c_{i-1}$.

Din (21), putem scrie

$$x_1 = y_1 - s_0,$$

(22)

$$x_i = y_i - y_{i-1} + c_{i-1}, \qquad i = 2,\ldots,6.$$

Atunci, cheltuielile totale, de cumpărare ale produsului, pe perioada de 6 luni, vor fi

$$(23) \quad \sum_{i=1}^{6} p_i x_i = -7y_1 + 5y_2 - 4y_3 - 3y_4 + 10y_5 + 10y_6 + 348$$

Este clar că y_i satisfac:

$$c_i \leqslant y_i \leqslant 9, \qquad i = 1,2,\ldots,5,$$

(24)

$$c_6 = y_6.$$

Condiția rezultă din limitarea stocului și din necesitatea acoperirii cererii lunare. Ultima egalitate exprimă necesitatea ca la sfîrșitul perioadei, stocul rămas să fie nul.

Totodată deoarece $x_i \geqslant 0$, avem și

$$y_1 - 2 \geqslant 0,$$

(25)

$$y_i - y_{i-1} + c_{i-1} \geqslant 0, \qquad i = 2, \ldots, 6.$$

Din (23), (24), (25), obținem următoarea formulare a problemei:

(26) $$\min\ (-7y_1 + 5y_2 - 4y_3 - 3y_4 + 10y_5 + y_6 + 348)$$

în condițiile

(27)
$$\begin{aligned} 8 &\leqslant y_1 \leqslant \min\,(9, y_2 + 8),\\ 5 &\leqslant y_2 \leqslant \min\,(9, y_3 + 5),\\ 3 &\leqslant y_3 \leqslant \min\,(9, y_4 + 3),\\ 2 &\leqslant y_4 \leqslant \min\,(9, y_5 + 2),\\ 7 &\leqslant y_5 \leqslant \min\,(9, y_6 + 7),\\ y_6 &= 4. \end{aligned}$$

Vom aplica procedeele programării dinamice, acestei probleme cu șase stadii. Întrucît valoarea ultimei variabile de control este cunoscută, etapele procesului vor inversa ordinea naturală.

Etapa I, Rezolvarea problemei

(I') $$\min_{y_6}\,[348 + 10y_6 + \min_{y_5,\ldots,y_1}\,(10y_5 - 3y_4 - 4y_3 + 5y_2 - 7y_1)]$$

este banală, singura valoare posibilă a lui y_6 fiind 4.

Deci $y_6^* = 4$.

Etapa a II-a. Corespunzător ultimelor două stadii, variabila y_5 și soluția optimă din etapa precedentă vor satisface

(II') $$\min\,[10y_5 + f_4]$$

Aici minimul se ia pentru valorile

$$7 \leqslant y_5 \leqslant \min (9, y_6 + 7).$$

Sau ţinînd seama de soluţia etapei I',

(28) $$7 \leqslant y_5 \leqslant 9,$$

(29) $$f_4 = \min [-3y_4 - 4y_3 + 5y_2 - 7y_1]$$

minimul fiind calculat în condiţiile corespunzătoare primelor patru stadii.

Ţinînd seama de (28), este uşor de văzut că soluţia problemei (II') este

$$y_5^* = 7.$$

Etapa a III-a. Corespunzător ultimilor trei stadii, variabilele y_4 şi y_5^*, y_6^* vor satisface

(III') $$\min [-3y_4 + f_3]$$

cu

$$2 \leqslant y_4 \leqslant \min (9, y_5 + 2)$$

adică, deoarece $y_5^* = 7$,

(30) $$2 \leqslant y_4 \leqslant 9.$$

De aici şi din (III')

$$y_4^* = 9.$$

Etapa a IV-a. Pentru determinarea lui f_3, luînd în considerare ultimele patru stadii, vom avea

(*IV'*) $$\min [-4y_3 + f_2]$$

unde

$$3 \leqslant y_3 \leqslant \min (9, y_4 + 3)$$

Scanned with OKEN Scanner

sau

$$3 \leqslant y_3 \leqslant 9.$$

Rezultă

$$y_3^* = 9.$$

Etapa a V-a. Pentru ultimele cinci stadii, variabila de control y_2 și soluția optimă y_3^*, y_3^*, y_4^*, y_3^*, rezultată din etapa precedentă optimizează problema

(V′) $\min\ [5y_2 + f_1]$

unde

$$5 \leqslant y_2 \leqslant \min\,(9, y_3 + 5)$$

sau

$$5 \leqslant y_2 \leqslant 9.$$

Este clar că, soluția problemei (*V′*) este dată de

$$y_2^* = 5.$$

Etapa a VI-a. Pentru toate cele șase stadii formulăm problema :

(VI′) $$\min(-7y_1)$$

unde

$$8 \leqslant y_1 \leqslant \min\,(9, y_2 + 8)$$

adică

$$8 \leqslant y_1 \leqslant 9.$$

De aici

$$y_1^* = 9.$$

În concluzie $y_1^* = 9$, $y_2^* = 5$, $y_3^* = 9$, $y_4^* = 9$, $y_5^* = 7$, $y_6^* = 4$, reprezintă o politică optimă în raport cu variabilele de control y_i $(i = 1, \ldots, 6)$.

Scanned with OKEN Scanner

Din (VI') avem

$$f_1 = -7y_1^* = -63.$$

Apoi din (V'), (IV'), (III'), (II'), (I'), vom avea succesiv:

$$f_2 = 5y_2^* + f_1 = -38,$$

$$f_3 = -4y_3^* + f_2 = -74,$$

$$f_4 = -3y_4^* + f_3 = -101.$$

$$f_5 = 10y_5^* + f_4 = -31,$$

$$z = 348 + 10y_6^* + f_5 = +357.$$

În sfîrşit, din (22) găsim

$$x_1^* = 7,\ x_2^* = 4,\ x_3^* = 9,\ x_4^* = 3,\ x_5^* = 0,\ x_6^* = 4.$$

Scanned with OKEN Scanner

BIBLIOGRAFIE

1. Bellman, R. E., Dreyfus, S. E., *Programare dinamică aplicată*, Editura tehnică Bucureşti, 1967.
2. Dantzig, G., *Applications et prolongements de la programmation lineaire* Dunod, Paris, 1966.
3. Dragomirescu, M., Maliţa, M., *Programarea pătratică*, Editura, ştiinţifică Bucureşti, 1968.
4. Gass, S., *Linear programming*, Mc Grow-Hill Book Co., 1969.
5. Hadley, G., *Linear programming*, Adison-Wesley Publishing Co.
6. Hadley, G., *Nonlinear and dynamic programming*, Adison-Wesley Publishing Co., 1964.
7. Kaufmann, A., Cruon, R., *La programmation dynamique*, Dunod, Paris, 1965.
8. Künzi, H. P., Krelle, W., *La programmation non lineaire*, Gauthier-Villars, Paris, 1969.
9. Mihoc, Gh., Ionescu, H., *Bazele matematice ale programării liniare*, Editura ştiinţifică, Bucureşti, 1965.
10. Mihoc, Gh., Nădejde Ileana, *Programarea matematică*, vol. I, II, Editura ştiinţifică, Bucureşti, 1966—1967.
11. Mihoc, Gh., Urseanu, V., *Matematici aplicate în statistică*, Editura Academiei R. S. R., 1962.
12. Simonnard, M., *Programmation lineaire*, Dunod, Paris, 1962.

Scanned with OKEN Scanner

CUPRINS

Prefață 3

Cap. I. NOȚIUNI MATEMATICE DE BAZĂ 7

§ 1. Algebră matriceală 7
§ 2. Rezolvarea sistemelor de ecuații liniare 22
§ 3. Spații vectoriale (reale) 27
§ 4. Mulțimi și funcții convexe 37
§ 5. Sisteme de inecuații liniare 42
§ 6. Forme liniare și pătratice 46
§ 7. Elemente de teoria probabilităților 49

Cap. II. PROGRAMAREA LINIARĂ 51

§ 1. Formularea problemei. Exemple 51
§ 2. Rezolvarea grafică a unor probleme simple 61
§ 3. Forma matematică a problemei 68
§ 4. Soluțiile problemei programării liniare 75
§ 5. Metoda simplex de rezolvare a programelor liniare 86
§ 6. Dualitatea în programarea liniară 115
§ 7. Algoritmul simplex dual 127
§ 8. Problema transporturilor 133

Cap. III. PROGRAMAREA NELINIARĂ 149

§ 1. Cazul general 149
§ 2. Programarea convexă. 172
§ 3. Programarea pătratică 189

Cap. IV. PROGRAMAREA DISCRETĂ. 200

§ 1. Probleme tipice aparținînd programării discrete 200
§ 2. Algoritmul lui Gomory (ciclic) pentru programarea liniară total discretă 209

Scanned with OKEN Scanner

§ 3. Rezolvarea programelor liniare mixte 218
§ 4. Programarea pătratică discretă 223

Cap. V. PROGRAMAREA LINIARĂ PARAMETRICĂ. 227

§ 1. Formularea problemei 227
§ 2. Programe liniare cu funcţia obiectiv parametrizată 230
§ 3. Parametrizarea vectorului b 238

Cap. VI. PROGRAMAREA LINIARĂ STOCHASTICĂ. 244

§ 1. Considerente generale 244
§ 2. Programarea liniară stochastică cu vectorul C aleator 250
§ 3. Programe stochastice cu vectorul b aleator 258

Cap. VII. PROGRAMAREA DINAMICĂ 265

§ 1. Principiul optimalităţii 265
§ 2. Tehnica de calcul a programării dinamice 267
§ 3. Rezolvarea unui exemplu numeric 270

BIBLIOGRAFIE 282

Coli de tipar: 17,75

Tiraj: 6 500+120 ex. leg.
Apărut: 1973

Întreprinderea Poligrafică Cluj
Str. Brassai nr. 5—7
Republica Socialistă România

Comanda nr. 83/5227

Scanned with OKEN Scanner